



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**MODERNÍ METODY POTLAČOVÁNÍ ŠUMU V
AUDIOSIGNÁLU ZALOŽENÉ NA FÁZI**

MODERN AUDIO DENOISING WITH UTILIZATION OF PHASE INFORMATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Pavel Skyva

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Mgr. Pavel Rajmic, Ph.D.

BRNO 2019

Diplomová práce

magisterský navazující studijní obor **Audio inženýrství**

Ústav telekomunikací

Student: Bc. Pavel Skyva

ID: 173739

Ročník: 2

Akademický rok: 2018/19

NÁZEV TÉMATU:

Moderní metody potlačování šumu v audiosignálu založené na fázi

POKYNY PRO VYPRACOVÁNÍ:

Diplomová práce se věnuje rekonstrukci audiosignálu, který byl zarušen nežádoucím šumem. Běžně metody odšumování spoléhají na informace uložené v magnitudě spektrogramu, ale ukazuje se, že fázová informace je taktéž důležitá. Student se seznámí s běžnými metodami pro řešení tohoto problému a také s přístupem, který využívá tzv. řídké reprezentace. Po nastudování teorie bude porovnáno několik modelů a příslušných algoritmů. Srovnání proběhne jak z hlediska výpočetní náročnosti algoritmů, tak z hlediska výstupů (tj. bude ohodnocena kvalita rekonstrukce signálu, a to jak subjektivně tak objektivně). Implementace proběhne Matlabu a bude ověřena na simulovaném i reálném signálu.

DOPORUČENÁ LITERATURA:

[1] Rajmic, P.; Daňková, M.: Úvod do řídkých reprezentací signálů a komprimovaného

snímání. Vysoké učení technické v Brně, 2014.

[2] Bayram, I.: Employing Phase Information for Audio Denoising, Proceedings of ICASSP, 2014.

Termín zadání: 1.2.2019

Termín odevzdání: 16.5.2019

Vedoucí práce: doc. Mgr. Pavel Rajmic, Ph.D.

Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

Tato diplomová práce se zabývá moderními metodami potlačování šumu ve zvukovém signálu. Rekonstrukce audiosignálu ze zašuměných dat je realizována především za využití fázové informace signálu a její derivace. K rekonstrukci se také využívá tzv. řídkých reprezentací signálů. V práci je popsán způsob hledání řídkého řešení pomocí proximálního Condatova algoritmu a jeho následné užití pro výpočet zrekonstruovaného signálu. Výpočetní program je implementován v prostředí MATLAB a k výpočtům využívá toolbox LTFAT. Výsledky rekonstrukce jsou hodnoceny objektivní metodou odstupu signálu od šumu (SNR) a stručně také pomocí subjektivního hodnocení.

KLÍČOVÁ SLOVA

Restaurace audiosignálu, fáze signálu, potlačování šumu, MATLAB, LTFAT, řídké reprezentace, Condatův algoritmus.

ABSTRACT

The thesis deals with modern methods of audio denoising. Reconstruction of the audiosignal is primarily based on utilization of phase information of signals and phase derivatives. Denoising methods also use sparse signal representations. In thesis is described the way of searching sparse coefficients using proximal Condat algorithm and following computation of reconstructed signal using this coefficients. The reconstruction algorithms are implemented in the MATLAB software with toolbox LTFAT included. Results of the reconstruction are compared using objective evaluation method Signal-to-Noise Ratio (SNR) and also by subjective evaluation.

KEYWORDS

Audio restoration, phase information, audio denoising, MATLAB, LTFAT, sparse representation, Condat algorithm.

SKYVA, Pavel. *Moderní metody potlačování šumu v audiosignálu založené na fázi*. Brno, 2019. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/118133>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Pavel Rajmic.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Moderní metody potlačování šumu v audiosignálu založené na fázi“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 16.5.2019

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval svému vedoucímu diplomové práce panu doc. Mgr. Pavlu Rajmicovi, Ph.D. za odborné vedení, věcné konzultace, vstřícný přístup a cenné rady. Dále bych rád poděkoval své rodině, že mě po celou dobu mého studia plně podporovala.

V Brně dne 16.5.2019

podpis autora

OBSAH

ÚVOD	8
1 ZNAČENÍ A ZÁKLADNÍ ZNALOSTI	9
1.1 NORMY VEKTORŮ	9
1.2 FROBENIOVA NORMA	10
1.3 DIFERENCE A DISKRÉTNÍ TOTÁLNÍ VARIACE.....	10
1.4 ŘÍDKOST.....	11
1.5 SKALÁRNÍ SOUČIN FUNKCÍ	11
2 BÁZE A FRAMY	12
2.1 VEKTOROVÝ PROSTOR.....	12
2.2 BÁZE VEKTOROVÉHO PROSTORU.....	12
2.2.1 <i>Ortogonální a ortonormální báze.....</i>	<i>13</i>
2.3 FRAMY.....	13
2.3.1 <i>Gaborovy framy.....</i>	<i>14</i>
3 FÁZOVÉ KONVENCE	16
3.1 STFT FÁZOVÉ KONVENCE	16
3.1.1 <i>Derivace fáze.....</i>	<i>17</i>
3.2 DGT FÁZOVÉ KONVENCE	17
3.2.1 <i>Derivace fáze.....</i>	<i>18</i>
4 ŘÍDKÉ REPREZENTACE SIGNÁLŮ	20
4.1 ALGORITMY PRO HLEDÁNÍ ŘÍDKÝCH ŘEŠENÍ.....	20
4.1.1 <i>ℓ_1-relaxace</i>	<i>21</i>
5 METODY ODŠUMOVÁNÍ SIGNÁLŮ	23
5.1 METODA ODŠUMOVÁNÍ SIGNÁLU ZALOŽENÁ NA INFORMACI ULOŽENÉ V MAGNITUDĚ SPEKTROGRAMU	23
5.2 MODERNÍ METODA ODŠUMOVÁNÍ ZALOŽENÁ NA FÁZI A VYUŽÍVAJÍCÍ ŘÍDKOSTI	24
5.3 MODERNÍ METODA ROZŠÍŘENÁ O DRUHOU DERIVACI A VERTIKÁLNÍ TOTÁLNÍ VARIACI	29
6 REALIZACE ALGORITMŮ V PROSTŘEDÍ MATLAB	32
7 TESTOVÁNÍ NAVRŽENÝCH ALGORITMŮ	38
7.1 HODNOCENÍ ÚSPĚŠNOSTI REKONSTRUKCE POMOCÍ SNR.....	38
7.2 TEST METODY VYUŽÍVAJÍCÍ MAGNITUDY SPEKTROGRAMU.....	38
7.3 TEST METODY ZALOŽENÉ NA PRVNÍ DERIVACI FÁZE A VYUŽÍVAJÍCÍ ŘÍDKOSTI	39
7.4 TEST METODY ZALOŽENÉ NA DRUHÉ DERIVACI FÁZE A VYUŽÍVAJÍCÍ ŘÍDKOSTI	40
7.5 ROZŠÍŘENÉ TESTOVÁNÍ REKONSTRUKČNÍCH METOD.....	42
ZÁVĚR	47
LITERATURA.....	48
SEZNAM SYMBOLŮ A ZKRATEK.....	51

SEZNAM PŘÍLOH	52
A OBSAH PŘILOŽENÉHO CD	53

ÚVOD

Tato diplomová práce se zabývá možnými způsoby rekonstrukce audiosignálu, který byl zarušen nežádoucím šumem. V dnešní době je kladen velký důraz na kvalitu zvukové reprodukce. Proto se často využívají různé nástroje, které by tuto kvalitu měly zlepšit. Jednou skupinou takových nástrojů jsou i nástroje na odstranění šumu ze zvukových nahrávek. Běžné metody odšumování spoléhají na informace uložené v magnitudě spektrogramu, ale ukazuje se, že fázová informace je taktéž důležitá. V práci si představíme jednu základní metodu odšumování vycházející právě z informací o magnitudě a formulujeme si také dvě moderní metody odšumování signálu využívající fázi signálu a tzv. řídké reprezentace signálů. Tyto metody pro nás budou v práci stěžejní. Všechny tři metody odstranění šumu implementujeme v prostředí MATLAB a provedeme jejich testování.

V prvních dvou kapitolách si stanovíme základní značení a definujeme si pojmy potřebné k porozumění dané problematice. Následovat bude kapitola zabývající se fází signálu a používanými fázovými konvencemi. Ve čtvrté kapitole si řekneme něco o řídkých reprezentacích signálů. Pátá kapitola již bude věnována samotnému problému rekonstrukce zašuměných audiosignálů pomocí již zmíněných metod. Popíšeme si metodu využívající informace uložené v magnitudě spektrogramu. Představíme si moderní metodu založenou na první derivaci fáze a využívající řídké reprezentace signálů. Tuto metodu následně rozšíříme o výpočet druhé derivace fáze. Šestá kapitola se bude věnovat již konkrétní realizaci navržených algoritmů v prostředí MATLAB. V poslední kapitole se zaměříme na testování metod, které byli v práci popsány. Výsledky testování budou porovnány především pomocí objektivní metody založené na odstupu signálu od šumu, ale také se pokusíme o stručné subjektivní hodnocení úspěšnosti rekonstrukce.

1 ZNAČENÍ A ZÁKLADNÍ ZNALOSTI

Skalární veličiny budeme značit kurzívou, např. takto m , N . Komplexní sdružení čísla z bude v textu označeno pruhem takto \bar{z} . Vektory budeme značit tučným řezem, jako např. \mathbf{x} , \mathbf{y} . Nebude-li řečeno jinak, pak budeme všechny konečněrozměrné vektory uvažovat jako sloupcové. Indexování prvků vektorů předpokládáme od jedničky, tzn. $\mathbf{x} = [x_1, \dots, x_n]^T$. Počet prvků množiny (tzv. kardinalitu) budeme značit stejně jako absolutní hodnotu, tedy např. $|\{2, -4, 8, 4, -1\}| = 5$.

Definice 1.1. *Nosičem vektoru \mathbf{x} rozumíme množinu jeho indexů, ve kterých má vektor nenulovou hodnotu. Tuto množinu značíme $\text{supp}(\mathbf{x})$. Platí tedy $\text{supp}(\mathbf{x}) = \{i | x_i \neq 0\}$.*

Matice budou značeny rovněž tučně, ale velkými písmeny, tj. \mathbf{A} , $\mathbf{\Psi}$. Jejich jednotlivé prvky pak příslušnými malými písmeny, tedy $a_{i,j}$, $\psi_{i,j}$. Řádek i matice \mathbf{A} bude označen symbolem $a_{i,:}$, obdobně sloupec j bude zastoupen symbolem $a_{:,j}$. Pokud nebude řečeno jinak, symbol $*$ bude označovat tzv. hermitovskou transpozici matice. Matice \mathbf{A}^* vznikne složením transpozice matice a komplexního sdružení každého jejího prvku, přičemž platí $(\mathbf{AB})^* = \mathbf{A}^* \mathbf{B}^*$. Operátorem $\text{vec}(\cdot)$ aplikovaným na matici budeme mít na mysli přeskládání všech jejích sloupců postupně pod sebe do jediného vektoru.

Vektorové prostory budeme značit takto: \mathbb{V} , \mathbb{R}^n , \mathbb{C}^n .

1.1 Normy vektorů

Norma vektoru je funkce, která jedním číslem vyjádří jeho velikost.

Definice 1.2. ℓ_p -norma vektoru $\mathbf{x} \in \mathbb{C}^N$ je definována takto

$$\begin{aligned} \|\mathbf{x}\|_p &:= \left(\sum_{i=1}^N |x_i|^p \right)^{\frac{1}{p}} \text{ pro } 1 \leq p < \infty, \\ \|\mathbf{x}\|_p &:= \sum_{i=1}^N |x_i|^p \text{ pro } 0 < p < 1, \\ \|\mathbf{x}\|_\infty &:= \max_i |x_i|, \\ \|\mathbf{x}\|_0 &:= |\text{supp}(\mathbf{x})|. \end{aligned} \tag{1.1}$$

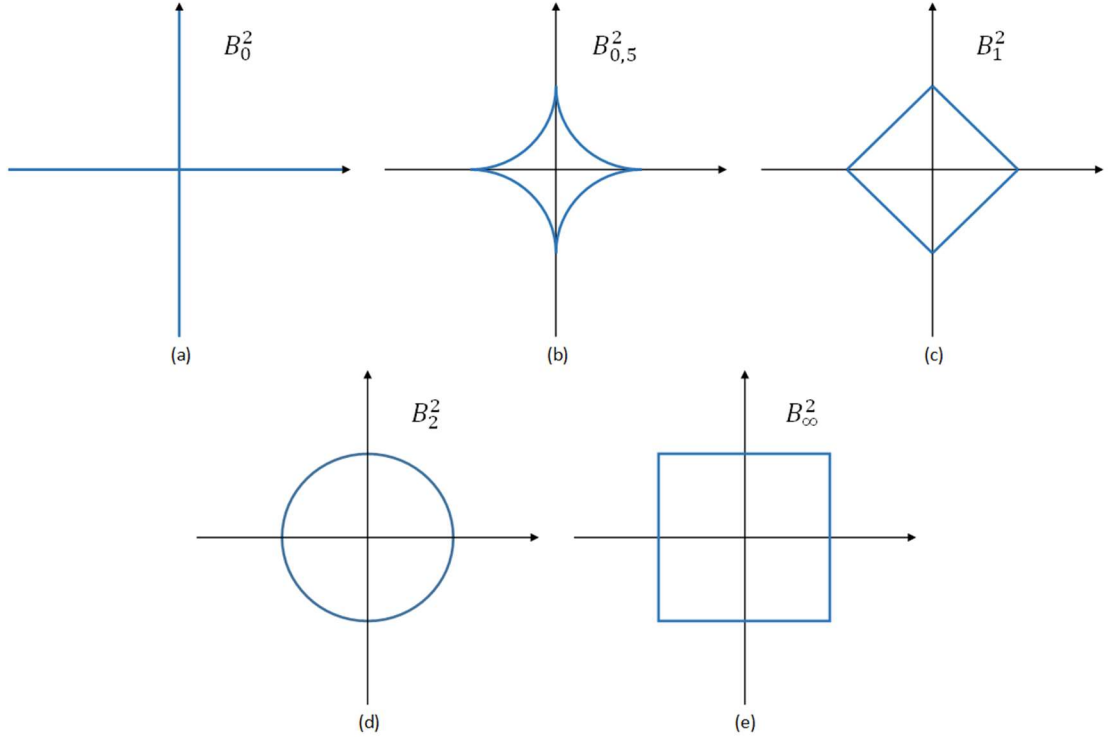
Nejčastěji se používají normy:

- $\|\cdot\|_0$, která představuje počet nenulových prvků vektoru, $\|\mathbf{x}\|_0 = |\text{supp}(\mathbf{x})|$,
- $\|\cdot\|_1$, která představuje součet absolutních hodnot prvků vektoru, $\|\mathbf{x}\|_1 = \sum_i |x_i|$,
- $\|\cdot\|_2$, která představuje tzv. eukleidovskou normu, $\|\mathbf{x}\|_2 = \sqrt{\sum_i |x_i|^2}$

V praxi je nejčastěji používaná právě eukleidovská norma, která se často značí zjednodušeně jako $\|\cdot\|$. Pro lepší představu si zobrazíme jednotkovou kouli v několika normách. Jejich zobrazení je na obrázku 1.1.

Definice 1.3. *Jednotková koule B_p^N je v normě ℓ_p definována jako*

$$B_p^N := \{\mathbf{x} \in \mathbb{C}^N | \|\mathbf{x}\|_p \leq 1\} \tag{1.2}$$



Obr. 1.1: Zobrazení hranic jednotkových koulí (a) B_0^2 , (b) $B_{0,5}^2$, (c) B_1^2 , (d) B_2^2 , (e) B_∞^2 .

1.2 Frobeniova norma

Definice 1.4. [1] Tzv. *Frobeniova norma* $\|\cdot\|_F$, která je nejběžnější maticovou normou, je definována jako $\|\mathbf{X}\|_F = \|\text{vec}(\mathbf{X})\|_2 = \sqrt{\sum_i \sum_j |x_{i,j}|^2}$. Jde tedy o energii prvků matice \mathbf{X} .

1.3 Diference a diskrétní totální variace

Diferenci budeme značit operátorem ∇ . Diference vektoru $\mathbf{x} \in \mathbb{C}^N$ se spočítají jako

$$\nabla \mathbf{x} = x_{i+1} - x_i. \quad (1.3)$$

Výsledný vektor bude mít délku $N - 1$.

Definice 1.5. Diskrétní totální variaci jednodimenzionálního vektoru $\|\mathbf{x}\|_{TV}$ definujeme jako

$$\|\mathbf{x}\|_{TV} := \sum_{i=1}^{n-1} |\nabla \mathbf{x}|. \quad (1.4)$$

Např. $\|[2, 0, -1, 4, 4, 1]\|_{TV} = |-2| + |-1| + |5| + |0| + |0| + |-3| = 11$.

Přestože diskrétní totální variace nesplňuje jeden z axiomů normy, v literatuře se běžně setkáváme s pojmem TV-norma [1].

1.4 Řídkost

Definice 1.6. Vektor $\mathbf{x} \in \mathbb{C}^N$ nazveme k -řídkým, pokud platí

$$\|\mathbf{x}\|_0 \leq k. \quad (1.5)$$

Vektor je tedy k -řídký, pokud obsahuje nejvýše k nenulových složek.

1.5 Skalární součin funkcí

Definice 1.7. Skalární součin spojitých funkcí $f(x)$ a $g(x)$ na uzavřeném intervalu $[a; b]$ definujeme jako

$$\langle f, g \rangle = \int_a^b f(x) \cdot \overline{g(x)} dx. \quad (1.6)$$

2 BÁZE A FRAMY

2.1 Vektorový prostor

Vektorovým prostorem rozumíme neprázdnou množinu \mathbb{V} , jejímiž prvky jsou vektory. Dimenze vektorového prostoru je rovna počtu prvků báze daného vektorového prostoru (více o bázích v části 2.2.). V našem případě se bude vždy jednat o vektorové prostory konečné dimenze. Dimenze tedy bude konečné číslo $0 < n \in \mathbb{N}$.

Definice 2.1. Máme-li vektorový prostor \mathbb{V} , potom pro každý vektor $\mathbf{x}, \mathbf{y} \in \mathbb{V}$ a každé $r, s \in \mathbb{R}$ platí

$$\begin{aligned}r \cdot (\mathbf{x} + \mathbf{y}) &= r \cdot \mathbf{x} + r \cdot \mathbf{y}, \\(r + s) \cdot \mathbf{x} &= r \cdot \mathbf{x} + s \cdot \mathbf{x}, \\(r \cdot s) \cdot \mathbf{x} &= r \cdot (s \cdot \mathbf{x}), \\1 \cdot \mathbf{x} &= \mathbf{x}.\end{aligned}\tag{2.1}$$

Vektor

$$\mathbf{x} = \sum_i \mathbf{x}_i r_i, r_i \in \mathbb{R}\tag{2.2}$$

nazveme lineární kombinací vektorů $\mathbf{x}_1, \dots, \mathbf{x}_n$.

Generátory vektorového prostoru \mathbb{V} rozumíme podmnožinu vektorů \mathbf{E} ve \mathbb{V} , která jej generuje. Tedy každý $\mathbf{x} \in \mathbb{V}$ lze vyjádřit nějakou lineární kombinací generátorů. Přičemž \mathbf{E} je matice tvořená generujícími vektory ve sloupcích vedle sebe. Vektorů generujících prostor \mathbb{V} může být i více než je dimenze daného prostoru. Vektor $\mathbf{x} \in \mathbb{V}$ potom můžeme vyjádřit vzájemně různými lineárními kombinacemi. Znamená to tedy, že tentýž vektor může mít více reprezentací. Tomu říkáme nedourčenost [1]. Vyjádříme-li \mathbf{x} pomocí jeho generátorů $\mathbf{E} = [\mathbf{e}_1 | \mathbf{e}_2 | \dots | \mathbf{e}_m]$ takto

$$\mathbf{x} = c_1 \mathbf{e}_1 + c_2 \mathbf{e}_2 + \dots + c_m \mathbf{e}_m = \mathbf{E}\mathbf{c},\tag{2.3}$$

pak skaláry c_i označujeme jako souřadnice \mathbf{x} v \mathbf{E} . To koresponduje s běžným způsobem modelování signálů, kdy je signál \mathbf{x} považován za superpozici dílčích signálů:

$$\mathbf{x} = \sum_{i=1}^m c_i \mathbf{e}_i,\tag{2.4}$$

kde i -tý koeficient odpovídá i -té souřadnici vektoru \mathbf{c} a i -tému sloupci matice \mathbf{E} . Z toho je patrné, že vztah (2.4) odpovídá vyjádření $\mathbf{x} = \mathbf{E}\mathbf{c}$ ve vztahu (2.3).

2.2 Báze vektorového prostoru

Jako bázi vektorového prostoru označujeme množinu lineárně nezávislých vektorů, jejichž lineární kombinací jsme schopni vyjádřit jakýkoliv vektor v daném vektorovém prostoru. Pro bázi $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ prostoru \mathbb{V} a každý vektor $\mathbf{x} \in \mathbb{V}$ pak platí

$$\mathbf{x} = \sum_{i=1}^n c_i \mathbf{b}_i = \mathbf{B}\mathbf{c}. \quad (2.5)$$

Jedná se tedy o jakýsi speciální případ množiny generátorů prostoru, kdy je na generující vektory kladen požadavek lineární nezávislosti. Získáme tak nejmenší počet generátorů, jaký je pro daný vektorový prostor možné získat. Dimenze vektorového prostoru je rovna počtu jeho bázevých vektorů. Má-li jeden prostor více bází, všechny mají stejný počet generujících vektorů.

2.2.1 Ortogonální a ortonormální báze

S ortogonálními a ortonormálními bázemi se v praxi pracuje nejjednodušeji. Pro ortogonální či ortonormální bázi platí, že jakékoliv dva její vektory jsou na sebe vždy kolmé. Potom tedy můžeme napsat, že pro libovolné dva vektory báze $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ platí

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0, \langle \mathbf{b}_i, \mathbf{b}_i \rangle \neq 0. \quad (2.6)$$

V ortonormální bázi navíc pro libovolný prvek \mathbf{b}_i platí $\|\mathbf{b}_i\| = 1$, tedy i $\mathbf{B}^* = \mathbf{B}^{-1}$.

2.3 Framy

Frame je rozšířením báze vektorového prostoru. Jak již bylo uvedeno v části 2.2, počet bázevých vektorů konečněrozměrného vektorového prostoru je roven dimenzi daného prostoru. Oproti tomu frame je tvořen množinou vektorů, jejichž počet může být i větší, než je dimenze prostoru. Pomocí těchto vektorů však stále můžeme vyjádřit jakýkoliv vektor daného vektorového prostoru. Z toho však vyplývá, že vektory tvořící frame již nutně musí být vzájemně lineárně závislé. Užití framů může mít výhody i nevýhody. Framy jsou méně omezené než báze, a proto se používají pro svoji flexibilitu. Nevýhodou mohou být náročnější výpočetní postupy a riziko numerické nestability [2].

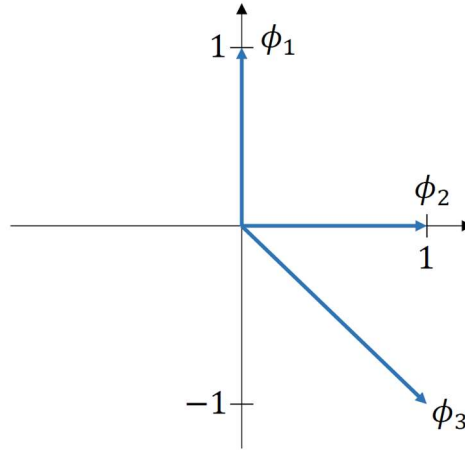
Definice 2.2. *Spočetnou množinu vektorů $\{\phi_k\}_{k \in \mathbb{I}}$ ve vektorovém prostoru \mathbb{V} nazveme frame, pokud existují takové konstanty $0 < A \leq B < \infty$, že platí*

$$A\|\mathbf{x}\|^2 \leq \sum_{k \in \mathbb{I}} |\langle \mathbf{x}, \phi_k \rangle|^2 \leq B\|\mathbf{x}\|^2, \forall \mathbf{x} \in \mathbb{V}. \quad (2.7)$$

Prvkům framu ϕ_k se obvykle říká atomy. Konstanty A, B se nazývají meze framu. Optimální dolní mez je supremum ze všech dolních mezí a optimální horní mez je infimum ze všech horních mezí. Pokud $A = B$, pak takový frame nazýváme těsný frame. Je-li dokonce $A = B = 1$, takový frame se nazývá 1-těsný frame nebo také těsný Parsevalův frame [1].

Na obrázku 2.1. vidíme příklad framu $\{\phi_1, \phi_2, \phi_3\}$ v \mathbb{R}^2 .

Generuje-li systém $\{\phi_k\}_{k \in \mathbb{I}}$ prostor \mathbb{V} , nemusí to ještě nutně znamenat, že ve \mathbb{V} tvoří frame. Např. systém $\{(1,0), (0,1), (0, \frac{1}{\sqrt{2}}), (0, \frac{1}{\sqrt{3}}), \dots\}$ určitě generuje \mathbb{R}^2 , ale konstanta B je nekonečná, protože výraz $\sum_{k \in \mathbb{I}} |\langle \mathbf{x}, \phi_k \rangle|^2 = |x_1|^2 + |x_2|^2 \cdot \sum_{k=1}^{\infty} \frac{1}{k}$ nekonverguje. Frame tedy není zobecněním báze, kde byla podmínka lineární nezávislosti prostě vypuštěna [1]. Uvažujeme-li však konečněrozměrný prostor dimenze n a konečnou množinu prvků $\{\phi_k\}_{k=1}^m$, $m \geq n$ generující



Obr. 2.1: Příklad framu v \mathbb{R}^2 .

\mathbb{V} , pak nám Cauchy-Schwarzova nerovnost [3]:

$$\sum_{k=1}^m |\langle \mathbf{x}, \phi_k \rangle|^2 \leq \sum_{k=1}^m \|\phi_k\|^2 \cdot \|\mathbf{x}\|^2, \forall \mathbf{x} \in \mathbb{V} \quad (2.8)$$

ukazuje, že konstanta B je konečné číslo. Podmínka horní meze je při uvedeném omezení tedy automaticky splněna. Stejně tak za uvedených podmínek existuje i dolní mez $A > 0$. Pokud by totiž nastala situace, že největší ze všech A je nulové, muselo by ve \mathbb{V} existovat nějaké $\mathbf{x} \neq 0$ takové, že suma v (2.7) je nulová a \mathbf{x} by tudíž bylo kolmé na celý prostor generovaný $\{\phi_k\}_{k \in \mathbb{I}}$, což je spor [1].

2.3.1 Gaborovy framy

Speciální skupinou framů jsou tzv. Gaborovy framy. Ty jsou dnes hojně využívány především pro časově-kmitočtovou analýzu.

Gaborova analýza v $L^2(\mathbb{R})$ je založena na dvou třídách operátorů v $L^2(\mathbb{R})$, jsou to:

- translace $T_a: L^2(\mathbb{R}) \rightarrow L^2(\mathbb{R})$, $(T_a f)(x) = f(x - a)$, $a \in \mathbb{R}$,
- modulace $E_b: L^2(\mathbb{R}) \rightarrow L^2(\mathbb{R})$, $(E_b f)(x) = f(x)e^{j2\pi b x}$, $b \in \mathbb{R}$,

kde a představuje translační parametr a b představuje modulační parametr. S využitím těchto operátorů se Gaborova analýza zaměřuje na reprezentaci funkce $f \in L^2(\mathbb{R})$ jako superpozici posunutých a modulovaných verzí fixní funkce $g \in L^2(\mathbb{R})$. Soubor takových funkcí

$$\{e^{j2\pi m b x} g(x - n a)\}_{m,n \in \mathbb{Z}} = \{E_{mb} T_{na} g\}_{m,n \in \mathbb{Z}} \quad (2.9)$$

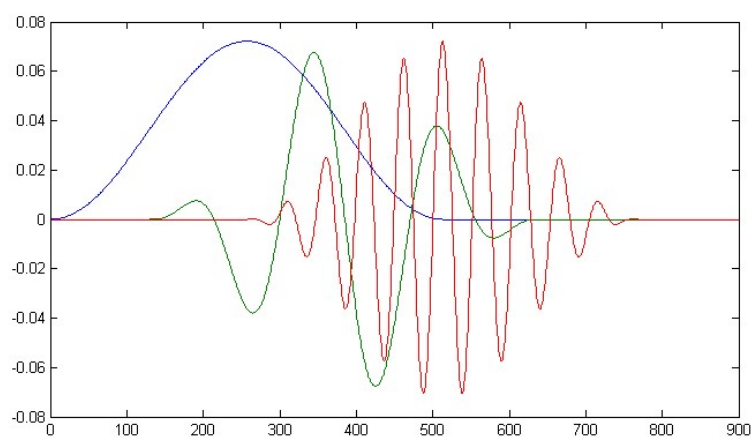
se pro zvolené dvě hodnoty a a b nazývá Gaborův (či gaborovský) systém, podle Dennise Gabora, který se jako první věnoval jejich matematické analýze [1]. Funkce E_{mb} reprezentuje modulované verze funkce g a funkce T_{na} představuje posunuté verze funkce g . Parametr na , $n \in \mathbb{Z}$ tak ovlivňuje pouze translaci, parametr mb , $m \in \mathbb{Z}$ ovlivňuje pouze modulaci. Tento přístup k reprezentaci signálů se často označuje jako STFT (krátkodobá Fourierova transformace) [4].

Základní otázkou je, jak zvolit funkci $g \in L^2(\mathbb{R})$ a parametry translace a modulae $a, b > 0$ tak, aby Gaborův systém (2.9) tvořil v prostoru $L^2(\mathbb{R})$ frame [2]. D. Gabor uvažoval posloupnost funkcí $\{E_{mb}T_{na}g\}_{m,n \in \mathbb{Z}}$, kde $ab = 1$ a g je Gaussova funkce

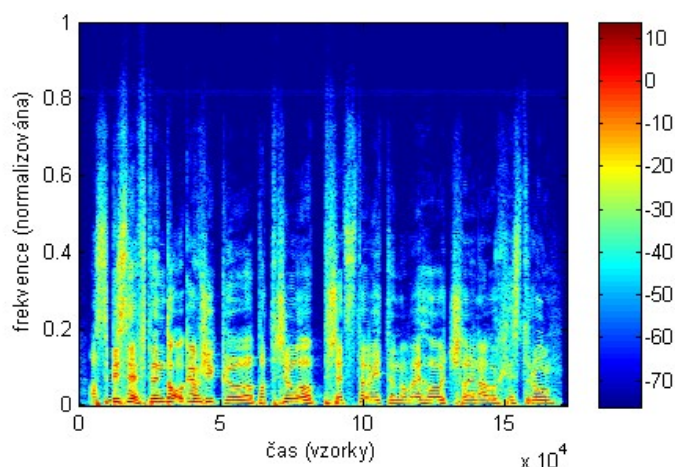
$$g(x) = e^{-\frac{x^2}{2}}. \quad (2.10)$$

Funkce g se nazývá okénková funkce [5]. Gaussova funkce jako jediná okénková funkce dosahuje optimální časově-kmitočtové koncentrace. Má však pro praxi nevýhodu nekonečně dlouhého nosiče [1]. Často proto na pozici okénkové funkce g vystupují jiná okna, jako např. Hammingovo, Hannovo nebo Tukeyovo. Na obrázku 2.2. můžeme vidět příklad posunutých a modulovaných Hannových oken.

Časové a frekvenční informace získané Gaborovou analýzou se často využívají k vystavění tzv. spektrogramu signálu. Většinou se jedná o dvojrozměrný obrázek s nepravými barvami. Příklad spektrogramu je na obrázku 2.3.



Obr. 2.2: Příklad posunutých a modulovaných okénkových funkcí tvořící Gaborův frame. Okénkovou funkcí je zde Hannovo okno (modře). Zobrazena je pouze reálná část.



Obr. 2.3: Příklad spektrogramu části reálného signálu.

3 FÁZOVÉ KONVENCE

V této kapitole se zaměříme na fázové konvence užívané ve výpočtech STFT a Diskrétní Gaborové transformaci (DGT) a jejich vliv na derivace fáze. Definujeme si pojmy jako je okamžitá frekvence nebo skupinové zpoždění, které souvisí s fázovými derivacemi.

3.1 STFT fázové konvence

STFT signálu $f(t)$ si za užití okénkové funkce $g(t)$ a fázové konvence (pc) dle [6] definujeme takto

$$V_{pc}(\omega, t) = M(\omega, t)e^{j\phi_{pc}(\omega, t)}. \quad (3.1)$$

Operátory translace a modulace vypadají následovně

$$T_\tau f(t) = f(t - \tau) \text{ a } M_\omega f(t) = f(t)e^{j\omega t}. \quad (3.2)$$

STFT s frekvenčně-invariantní fází je poté definována jako

$$V_{fi}(\omega, t) = \langle f, M_\omega T_t g \rangle. \quad (3.3)$$

S využitím definice 1.7. potom získáme

$$V_{fi}(\omega, t) = \int_{-\infty}^{\infty} f(\tau)e^{-j\omega\tau}\overline{g(\tau - t)}d\tau. \quad (3.4)$$

STFT s časově-invariantní fází vypadá následovně

$$\begin{aligned} V_{ti}(\omega, t) &= \langle f, T_t M_\omega g \rangle = \\ &= \int_{-\infty}^{\infty} f(\tau)e^{-j\omega(\tau - t)}\overline{g(\tau - t)}d\tau = \\ &= \int_{-\infty}^{\infty} f(\tau + t)e^{-j\omega\tau}\overline{g(\tau)}d\tau = \\ &= e^{j\omega t}V_{fi}(\omega, t). \end{aligned} \quad (3.5)$$

STFT se symetrickou fází je definována jako

$$\begin{aligned} V_{sym}(\omega, t) &= \langle f, M_{\omega/2} T_t M_{\omega/2} g \rangle = \\ &= \int_{-\infty}^{\infty} f(\tau)e^{-j\omega\tau/2}e^{-j\omega(\tau - t)/2}\overline{g(\tau - t)}d\tau = \\ &= \int_{-\infty}^{\infty} f(\tau)e^{-j\omega(\tau - t/2)}\overline{g(\tau - t)}d\tau = \\ &= \int_{-\infty}^{\infty} f(\tau + t/2)e^{-j\omega\tau}\overline{g(\tau - t/2)}d\tau = \\ &= e^{j\omega t/2}V_{fi}(\omega, t). \end{aligned} \quad (3.6)$$

Ze vztahů (3.4), (3.5) a (3.6) poté získáme jednotlivé vztahy mezi fázemi, které jsou následující

$$\phi_{fi}(\omega, t) = \phi_{ti}(\omega, t) - \omega t = \phi_{sym}(\omega, t) - \omega t/2. \quad (3.7)$$

3.1.1 Derivace fáze

Derivace fáze podle frekvence vypadá následovně

$$\frac{\partial \phi_{fi}}{\partial \omega} = \frac{\partial \phi_{ti}}{\partial \omega} - t = \frac{\partial \phi_{sym}}{\partial \omega} - \frac{t}{2}. \quad (3.8)$$

Obdobně derivace fáze podle času je následující

$$\frac{\partial \phi_{fi}}{\partial t} = \frac{\partial \phi_{ti}}{\partial t} - \omega = \frac{\partial \phi_{sym}}{\partial t} - \frac{\omega}{2}. \quad (3.9)$$

Derivace $\frac{\partial \phi_{ti}}{\partial t}$ bývá často označována jako okamžitá frekvence $\hat{\omega}(\omega, t)$,

$$\hat{\omega}(\omega, t) = \frac{\partial \phi_{ti}}{\partial t} = \omega + \frac{\partial \phi_{fi}}{\partial t} = \frac{\omega}{2} + \frac{\partial \phi_{sym}}{\partial t}. \quad (3.10)$$

Druhá derivace fáze již nezávisí na použité fázové konvenci. Tedy

$$\frac{\partial^2 \phi_{ti}}{\partial \omega^2} = \frac{\partial^2 \phi_{fi}}{\partial \omega^2} = \frac{\partial^2 \phi_{sym}}{\partial \omega^2}, \quad (3.11)$$

$$\frac{\partial^2 \phi_{ti}}{\partial t^2} = \frac{\partial^2 \phi_{fi}}{\partial t^2} = \frac{\partial^2 \phi_{sym}}{\partial t^2}. \quad (3.12)$$

Skupinové zpoždění (local group delay, LGD) definuje [6] jako

$$\text{LGD}(\omega, t) = -\frac{\partial \phi_{ti}}{\partial \omega}(\omega, t). \quad (3.13)$$

3.2 DGT fázové konvence

DGT signálu $f \in \mathbb{C}^L$ si za užití okénkové funkce g a fázové konvence (pc) dle [6] definujeme jako

$$c_{pc}(m, n) = s(m, n)e^{j\varphi_{pc}(m, n)}, \quad (3.14)$$

kde m a n jsou frekvenční a časové indexy.

DGT s frekvenčně-invariantní fází je následující

$$\begin{aligned} c_{fi}(m, n) &= \langle f, M_{mb}T_{na}g \rangle = \\ &= \sum_{l=0}^{L-1} f(l)e^{-j2\pi mbl/L} \overline{g(l-an)} \end{aligned} \quad (3.15)$$

pro $m = 0, \dots, M - 1$ a $n = 0, \dots, N - 1$, kde $M = L/b$ je počet kanálů (modulací). $N = L/a$ je počet posunutí. Parametr a je potom posun v čase a parametr b představuje frekvenční rozlišení.

DGT s časově-invariantní fází je definována jako

$$\begin{aligned} c_{\text{ti}}(m, n) &= \langle f, T_{na} M_{mb} g \rangle = \\ &= \sum_{l=0}^{L-1} f(l) e^{-j2\pi mb(l-an)/L} \overline{g(l-an)} = \\ &= \sum_{l=0}^{L-1} f(l+an) e^{-j2\pi mb/l/L} \overline{g(l)} = \\ &= e^{j2\pi mbna/L} c_{\text{fi}}(m, n). \end{aligned} \quad (3.16)$$

DGT se symetrickou fází vypadá následovně

$$\begin{aligned} c_{\text{sym}}(m, n) &= \langle f, M_{mb/2} T_{na} M_{mb/2} g \rangle = \\ &= \sum_{l=0}^{L-1} f(l) e^{-j2\pi mb(l-an/2)/L} \overline{g(l-an)} = \\ &= e^{j\pi mbna/L} c_{\text{fi}}(m, n). \end{aligned} \quad (3.17)$$

Je-li při užití symetrické fáze b liché, dochází k nespojitosti fáze.

Vzájemné vztahy mezi fázemi jsou

$$\varphi_{\text{fi}}(m, n) = \varphi_{\text{ti}}(m, n) - 2\pi mbna/L = \varphi_{\text{sym}}(m, n) - \pi mbna/L. \quad (3.18)$$

3.2.1 Derivace fáze

V případě diskretních signálů jsme schopni vyjádřit derivace fáze pouze numericky určitou aproximací [6]. Tyto derivace budeme poněkud zjednodušeně značit $\frac{\partial \varphi(m, n)}{\partial \omega}$ a $\frac{\partial \varphi(m, n)}{\partial t}$.

Derivace fáze podle frekvence má následující vyjádření

$$\frac{\partial \varphi_{\text{fi}}}{\partial \omega} = \frac{\partial \varphi_{\text{ti}}}{\partial \omega} - na = \frac{\partial \varphi_{\text{sym}}}{\partial \omega} - \frac{na}{2}. \quad (3.19)$$

Obdobně derivace fáze podle času je následující

$$\hat{m}(m, n)b = \frac{\partial \varphi_{\text{ti}}}{\partial t} = bm + \frac{\partial \varphi_{\text{fi}}}{\partial t} = \frac{bm}{2} + \frac{\partial \varphi_{\text{sym}}}{\partial t}, \quad (3.20)$$

kde $\hat{m}(m, n)$ představuje okamžitý frekvenční index v reprezentaci signálu.

Druhá derivace již opět nezávisí na použité fázové konvenci, tedy

$$\frac{\partial^2 \varphi_{\text{fi}}}{\partial \omega^2} = \frac{\partial^2 \varphi_{\text{ti}}}{\partial \omega^2} = \frac{\partial^2 \varphi_{\text{sym}}}{\partial \omega^2}, \quad (3.21)$$

$$\frac{\partial^2 \varphi_{\text{fi}}}{\partial t^2} = \frac{\partial^2 \varphi_{\text{ti}}}{\partial t^2} = \frac{\partial^2 \varphi_{\text{sym}}}{\partial t^2}. \quad (3.22)$$

4 ŘÍDKÉ REPREZENTACE SIGNÁLŮ

Při řešení nedourčených systémů lineárních rovnic pro nás mohou ze všech možných řešení být výhodná taková, která jsou tzv. řídká (mají co nejvíce neznámých současně nulových). Mohou nám usnadnit interpretaci dat, napomoci snadné a silné kompresi, případně v množství dalších aplikací [1]. Podíváme-li se například na ukázkou spektrogramu na obrázku 2.3, vidíme, že značná část Gaborových koeficientů má úroveň menší než -40 dB, což jsou hodnoty téměř zanedbatelné vůči ostatním koeficientům. Podstatnou informaci tedy u řady reálných signálů nese pouze část koeficientů.

Budeme-li řešit soustavu lineárních rovnic $\mathbf{y} = \mathbf{Ax}$, pak máme zájem takový, aby $\|\mathbf{x}\|_0$ bylo co nejmenší. Tento problém bychom mohli formulovat takto

$$\mathbf{x} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k } \mathbf{y} = \mathbf{Ax}, \quad (4.1)$$

kde známe vektor $\mathbf{y} \in \mathbb{C}^m$ a matici $\mathbf{A} \in \mathbb{C}^{N \times M}$, pro kterou platí $M > N$ (obvykle $M \gg N$). Matice \mathbf{A} se nazývá slovník, jejím sloupcům se obvykle říká atomy [7]. $\|\mathbf{x}\|_0$ je ℓ_0 -norma rovna počtu nenulových prvků vektoru \mathbf{x} :

$$\|\mathbf{x}\|_0 := |\text{supp}(\mathbf{x})|. \quad (4.2)$$

Výpočetní složitost problému (4.1) odpovídá složitosti třídy NP. To znamená, že výsledku je možné dosáhnout v čase vyšším než polynomiálním [8], což je pro nás nepřijatelné. Pro získání výsledku v rozumném čase musíme přistoupit k některé z aproximativních metod. Užití aproximativních metod nás však vystavuje riziku úbytku přesnosti.

4.1 Algoritmy pro hledání řídkých řešení

Podle principu na kterém se zakládají, lze rozdělit algoritmy pro hledání řídkých řešení do dvou skupin. Do první skupiny řadíme metody nazvané jako relaxační. Ty vychází z tzv. ℓ_1 -relaxace. Druhou skupinu tvoří tzv. hladové („greedy“) algoritmy [7].

Mezi relaxační algoritmy řadíme například:

- BP (Basis Pursuit) [9]
- Modifikovaný LARS (Least Angle Regression, homotopy method) [10]
- IRLS (Iterative Reweighted Least Squares, někdy také jako FOCUSS – FOCal Underdetermined System Solver) [11]
- Dantzig Selector [12]

Za dodržení určitých podmínek ℓ_1 -relaxace (viz část 4.1.1), jsou tyto metody schopné dostat se k přesnému nebo alespoň přibližnému řešení.

Jako hladové algoritmy potom označujeme například:

- MP (Matching Pursuit) [13]
- OMP (Orthogonal Matching Pursuit) [14]

Princip těchto algoritmů spočívá v tom, že v každé iteraci najdou jeden (nebo více) „nejvýznamnějších“ atomů, který (případně které) se v dalším průběhu algoritmu už nemohou

zbavit podílu na konečném řešení. Výhodou těchto metod je nízká složitost, nevýhodou pak, že není zaručeno dosažení globálního optima [7].

Mimo tyto dvě skupiny algoritmů stojí ještě algoritmy založené na prahování (thresholdingu) [15] a tzv. hybridní algoritmy, které využívají jednotlivé výhody z obou skupin algoritmů.

4.1.1 ℓ_1 -relaxace

Abychom byli schopni problém (4.1) řešit pomocí algoritmů konvexní optimalizace, přistupujeme k jeho relaxaci. Vzhledem k tomu, že normy ℓ_p jsou konvexní pro $p \geq 1$, jako „nejbližší“ konvexní norma k normě ℓ_0 se nám nabízí ℓ_1 -norma. ℓ_1 -relaxací bychom tak měli být schopni dosáhnout přesného nebo alespoň přibližného řešení problému. Relaxovaný problém (4.1) poté vypadá následovně

$$\mathbf{x} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhledem k } \mathbf{y} = \mathbf{Ax}. \quad (4.3)$$

V některých případech se řešení problémů (4.1) a (4.3) dokonce shodují. Uvažujeme-li však možnou odchylku řešení δ (zašuměná data/nepřesnosti), daný problém bychom mohli napsat jako

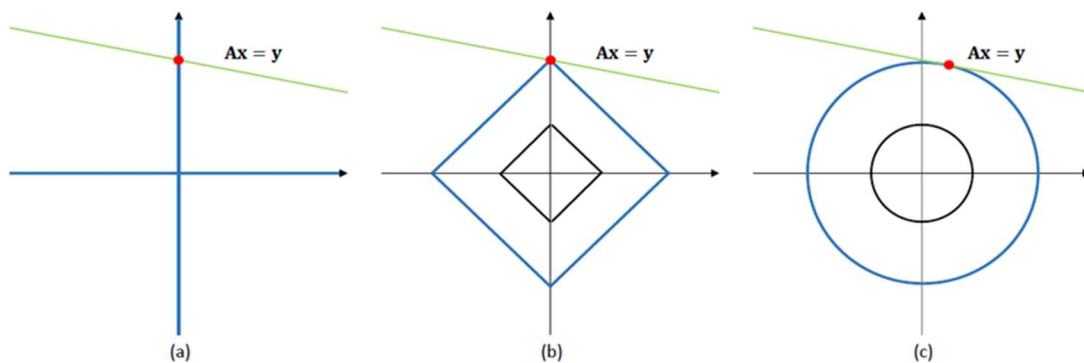
$$\mathbf{x} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhledem k } \|\mathbf{Ax} - \mathbf{y}\|_2 \leq \delta. \quad (4.3\delta)$$

Problém (3.3 δ) je také znám pod zkratkou LASSO (Least Absolute Shrinkage and Selection Operator) [10].

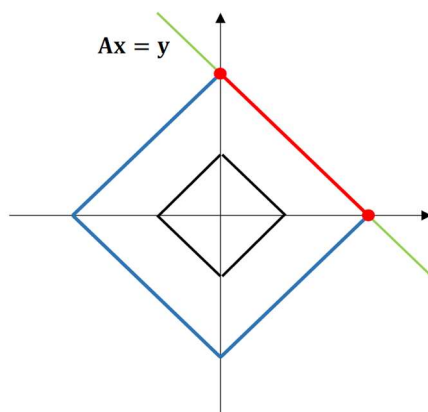
Na obrázku 4.1 je znázorněno řešení úlohy

$$\mathbf{x} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_p \text{ vzhledem k } \mathbf{y} = \mathbf{Ax} \quad (4.4)$$

v \mathbb{R}^2 jednotlivě pro $p = 0; 1; 2$. Všechna přípustná řešení zde znázorňuje zelená přímka. Z vyobrazení je patrné, že řešení (4.1) a (4.3) jsou shodná. Je zde však také vidět, že tomu tak nemusí být vždy. Pokud by přímka všech přípustných řešení $\mathbf{y} = \mathbf{Ax}$ byla rovnoběžná s hranou jednotkové kružnice normy ℓ_1 , získali bychom nekonečně mnoho řešení, přičemž jen ta krajní by byla řídká. Tuto situaci znázorňuje obrázek 4.2.



Obr. 4.1: „Nafukující se“ koule v normách (a) ℓ_0 , (b) ℓ_1 a (c) ℓ_2 při styku s nadrovinou určenou rovnicí $\mathbf{Ax} = \mathbf{y}$.



Obr. 4.2: Ukázka selhání ℓ_1 -relaxace při hledání řídkého řešení v \mathbb{R}^2 .

5 METODY ODŠUMOVÁNÍ SIGNÁLŮ

K dispozici budeme mít monofonní audiosignál $\mathbf{y} \in \mathbb{R}^N$, který však bude zarušen nežádoucím šumem. V našem případě se bude vždy jednat o Gaussův bílý šum. Jedná se tedy o šum s normálním rozdělením, jehož výkon je stejný v libovolných frekvenčních pásmech stejné šířky. Cílem tedy bude, pokusit se zrekonstruovat audio signál z jeho zašuměného pozorování.

V této kapitole si nejprve představíme jednu běžnou metodu odšumování signálu spoléhající na informace uložené v magnitudě spektrogramu. Dále se zaměříme na metodu odstranění šumu za pomoci moderního způsobu založeného na řídkých reprezentacích a využívající informaci o fázi získané z časově-frekvenční reprezentace signálu. Tato metoda pro nás bude v této práci stěžejní.

5.1 Metoda odšumování signálu založená na informaci uložené v magnitudě spektrogramu

Běžné metody odšumování spoléhají na informace uložené v magnitudě spektrogramu. Spočívají v extrakci magnitudy a fáze z koeficientů časově-frekvenční reprezentace. Informace o magnitudě je potom zpracovávána zvlášť. Princip odstranění šumu je založen na útlumu malých hodnot magnitudy v reprezentaci, zatímco velké hodnoty se ponechají beze změny. Předpokladem úspěšného odšumění signálu touto metodou tedy je, že úroveň šumu je výrazně menší než úroveň požadovaného signálu. Fázová informace signálu se ponechává beze změny.

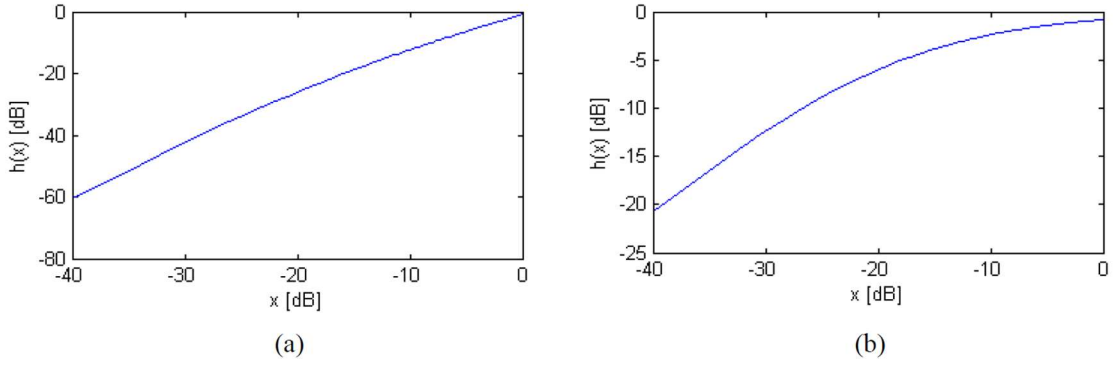
Máme-li signál $\mathbf{y} \in \mathbb{R}^N$ s jeho časově-frekvenční reprezentací $c(m, n) = s(m, n)e^{j\varphi(m, n)}$, kde m jsou frekvenční indexy a n jsou časové indexy, je s magnitudami $s(m, n)$ provedena nelineární operace

$$s_d(m, n) = h(s(m, n)), \quad (5.1)$$

$s_d(m, n)$ představuje matici magnitud odšuměného signálu. Výstupem jsou tedy časově-frekvenční koeficienty odšuměného signálu $\hat{z}(m, n) = s_d(m, n)e^{j\varphi(m, n)}$, jejichž převedením zpět do časové oblasti získáme signál zbavený šumu.

Nelineární funkci představuje funkce $h(x)$. Běžným příkladem takové funkce je $h(x) = x^2/(x + \eta)$ [16]. Další používanou podobou nelineární funkce je také $h(x) = x/(x + \eta)$. Tyto funkce můžeme vidět na obrázku 5.1. Konstanta η udává míru útlumu.

Přestože by tato metoda měla být schopná poměrně dobře odstranit šum z tichých pasáží signálu, její velkou nevýhodou zůstává to, že šum namodulovaný na frekvenčních složkách s velkou magnitudou zůstane nepostihnut.



Obr. 5.1: Nelineární funkce (a) $h(x) = x^2/(x + \eta)$ a (b) $h(x) = x/(x + \eta)$ používané pro útlum magnitud spektrogramu.

5.2 Moderní metoda odšumování založená na fázi a využívající řídkosti

Máme k dispozici vstupní signál $\mathbf{y} \in \mathbb{R}^N$ znehodnocený šumem s jeho diskrétní časově-frekvenční reprezentací $c(m, n) = s(m, n)e^{j\varphi(m, n)}$, kde m jsou frekvenční indexy a n jsou časové indexy.

Článek [17] definuje upravené $c(m, n)$ jako

$$\hat{c}(m, n) = |c(m, n)| \frac{c(m, n+1)}{|c(m, n+1)|} = |c(m, n)| e^{j\arg(c(m, n+1))}. \quad (5.2)$$

Budeme hledat takové $\alpha(m, n)$, pro které platí

$$c(m, n) \approx \hat{c}(m, n)\alpha(m, n). \quad (5.3)$$

V tom případě by nejlepší $\alpha(m, n)$ vypadalo takto

$$\alpha(m, n) = \frac{c(m, n)}{\hat{c}(m, n)} = \frac{|c(m, n)|}{|\hat{c}(m, n)|} e^{j\arg(c(m, n)) - j\arg(\hat{c}(m, n))}, \quad (5.4)$$

přičemž ze vztahů (5.2) a (5.4) vyplývá, že

$$|\alpha(m, n)| \leq 1 \quad (5.5)$$

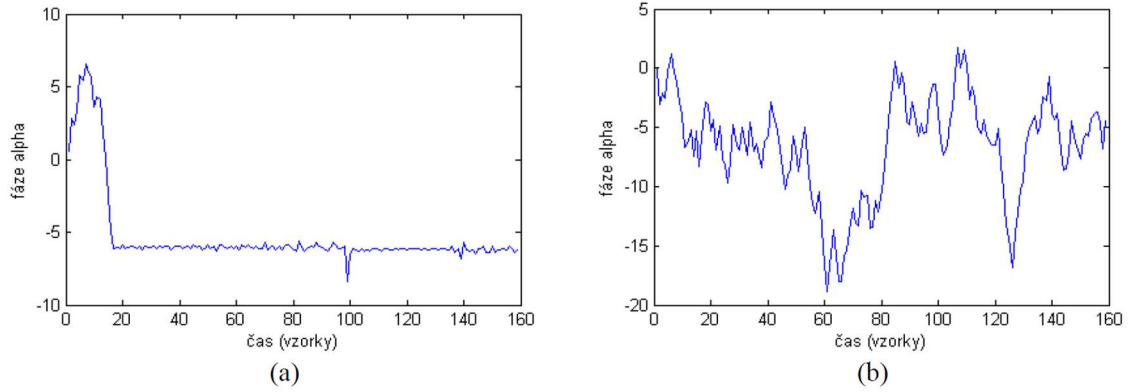
a také

$$\arg(\alpha(m, n)) = \arg(c(m, n)) - \arg(c(m, n+1)), \quad (5.6)$$

což pro nás představuje „derivaci“ fáze podle času. Na obrázku 5.2 je zobrazena fáze jednoho řádku $\alpha(m, :)$ jednotlivě pro oblast s výskytem harmonických komponentů vstupního signálu a pro oblast bez výskytu harmonických komponentů (tedy oblast s převahou šumu).

Naším cílem je získat časově-frekvenční reprezentaci odšuměného signálu

$$z(m, n) = \hat{c}(m, n)\hat{\alpha}(m, n). \quad (5.7)$$



Obr. 5.2: Zobrazení fáze jednoho řádku $\alpha(m, :)$ jednotlivě pro (a) oblast s výskytem harmonických komponentů vstupního signálu a pro (b) oblast bez výskytu harmonických komponentů (tedy oblast s převahou šumu).

Aby bylo zřejmé, že pracujeme s mapami koeficientů, $c(m, n)$ budeme označovat \mathbf{C} , obdobně $\hat{c}(m, n)$ jako $\hat{\mathbf{C}}$, řádkovou mapu $\alpha(m, n)$ označíme \mathbf{A} a odhad $\hat{\alpha}(m, n)$ jako $\hat{\mathbf{A}}$. Abychom získali vhodné $\hat{\mathbf{A}}$, řešíme následující minimalizační úlohu

$$\hat{\mathbf{A}} = \arg \min_{\|\mathbf{A}\|_1 \leq 1} \frac{1}{2} \|\mathbf{C} - \hat{\mathbf{C}}\mathbf{A}\|_F^2 + \lambda_{TV} \text{TV}(\mathbf{A}) + \lambda_1 \|\mathbf{A}\|_1, \quad (5.8)$$

ve které na $\hat{\mathbf{A}}$ klademe několik požadavků. Jedním z nich je, aby se nové časově-frekvenční koeficienty příliš nelišily od původních, to zajišťuje člen $\|\mathbf{C} - \hat{\mathbf{C}}\mathbf{A}\|_F^2/2$. Dále zde jsou dva regularizační členy, jimž je v minimalizaci přikládána váha dle konstant λ_{TV} a λ_1 . Nastavení λ_{TV} a λ_1 také určuje míru, s jakou upřednostňujeme tyto regularizační členy před požadavkem $\|\mathbf{C} - \hat{\mathbf{C}}\mathbf{A}\|_F^2/2$. Člen $\text{TV}(\mathbf{A})$ v úloze představuje totální variaci \mathbf{A} po řádcích (ve směru časové osy). Předpokládáme-li totiž signál se sinusoidálními komponenty, pak by \mathbf{A} pro tyto komponenty mělo být po řádcích konstantní. Pro tyto komponenty bude tedy malá i řádková totální variace. Tento předpoklad dokládá obrázek 5.2. Člen $\|\mathbf{A}\|_1$ v úloze (5.8) vynucuje řádkovost \mathbf{A} . Významné časově-frekvenční koeficienty signálu budou zachovány, zatímco méně významné koeficienty budou potlačeny.

V následujících řádcích si úlohu (5.8) přepíšeme do podoby vhodné pro použití proximálního Condatova algoritmu [19].

$$\arg \min_{\|\mathbf{A}\|_\infty \leq 1} \frac{1}{2} \|\mathbf{C} - \hat{\mathbf{C}}\mathbf{A}\|_F^2 + \lambda_{TV} \text{TV}(\mathbf{A}) + \lambda_1 \|\mathbf{A}\|_1 = \quad (5.9)$$

$$= \arg \min_{\|\mathbf{A}\|_\infty \leq 1} \frac{1}{2} \|\mathbf{C} - \hat{\mathbf{C}}\mathbf{A}\|_F^2 + \|\lambda_{TV} L_{TV} \mathbf{A}\|_1 + \|\lambda_1 \mathbf{A}\|_1 = \quad (5.10)$$

$$= \arg \min_{\|\mathbf{A}\|_\infty \leq 1} \frac{1}{2} \|\mathbf{C} - \hat{\mathbf{C}}\mathbf{A}\|_F^2 + \left\| \begin{bmatrix} \lambda_{TV} L_{TV} \\ \lambda_1 \mathbf{I} \end{bmatrix} \mathbf{A} \right\|_1 = \quad (5.11)$$

$$= \arg \min_{\mathbf{A}} \frac{1}{2} \|\mathbf{C} - \hat{\mathbf{C}}\mathbf{A}\|_F^2 + \left\| \begin{bmatrix} \lambda_{TV} L_{TV} \\ \lambda_1 \mathbf{I} \end{bmatrix} \mathbf{A} \right\|_1 + \iota_{\{\mathbf{Z} | \|\mathbf{Z}\|_\infty \leq 1\}}(\mathbf{A}). \quad (5.12)$$

Operátor \odot představuje násobení matic po jednotlivých prvcích. Matice \mathbf{I} představuje identitu (jednotkovou matici). Pro indikátorovou funkci $\iota_{\{Z||Z||_\infty \leq 1\}}(\mathbf{A})$ platí:

$$\iota_{\{Z||Z||_\infty \leq 1\}}(\mathbf{A}) = \begin{cases} 0, & \text{pokud } \|\mathbf{A}\|_\infty \leq 1, \\ \infty, & \text{pokud } \|\mathbf{A}\|_\infty \not\leq 1. \end{cases} \quad (5.13)$$

Článek [18] uvádí, že proximální Condatův algoritmus lze aplikovat na úlohy v podobě

$$\arg \min_x f(x) + g(x) + h(Lx), \quad (5.14)$$

se všemi funkcemi konvexními, funkcí f diferencovatelnou a lineárním operátorem L . V našem případě by potom $f = \|\mathbf{C} - \hat{\mathbf{C}} \odot \cdot\|_{\mathbb{F}}^2/2$, $g = \iota_{\{Z||Z||_\infty \leq 1\}}$, $h = \|\cdot\|_1$ a $L = \begin{bmatrix} \lambda_{\text{TV}} L_{\text{TV}} \\ \lambda_1 \mathbf{I} \end{bmatrix}$.

Konkrétní podoba Condatova algoritmu řešící (5.12) je dle [19] následující:

Algoritmus 5.1.

Vstupem jsou \mathbf{C} , $\hat{\mathbf{C}}$, λ_1 , λ_{TV} a lineární operátor L_{TV} .

Výstupem bude $x^{(i+1)}$.

Nastaví se parametry $\tau, \sigma > 0$;

nastaví se proměnná $x^{(0)}$ a duální proměnné $u_{\text{TV}}^{(0)}$, $u_1^{(0)}$;

vytvoří se čítač iterací $i = 0$.

V každé iteraci provede algoritmus výpočty:

$$x^{(i+1)} = \text{clip}_1 \left(x^{(i)} - \tau \left[|\hat{\mathbf{C}}|^2 \odot x^{(i)} - \hat{\mathbf{C}} \odot \mathbf{C} + \lambda_{\text{TV}} L_{\text{TV}}^* u_{\text{TV}}^{(i)} + \lambda_1 u_1^{(i)} \right] \right);$$

$$p = 2x^{(i+1)} - x^{(i)};$$

$$q_{\text{TV}} = u_{\text{TV}}^{(i)} + \sigma \lambda_{\text{TV}} L_{\text{TV}} p;$$

$$q_1 = u_1^{(i)} + \sigma \lambda_1 p;$$

$$u_{\text{TV}}^{(i+1)} = \text{clip}_1(q_{\text{TV}});$$

$$u_1^{(i+1)} = \text{clip}_1(q_1);$$

$$i = i + 1.$$

Algoritmus každou iterací konverguje k jeho řešení. Iterace ukončíme, je-li splněno nějaké vhodné kritérium.

Výstup $x^{(i+1)}$ algoritmu 5.1. poté odpovídá hledanému $\hat{a}(m, n)$, které použijeme pro získání časově-frekvenční reprezentace $z(m, n) = \hat{c}(m, n) \hat{a}(m, n)$ odšuměného signálu. Ukázku výsledné řídké mapy $\hat{a}(m, n)$ můžeme vidět na obrázku 5.4 (c).

Lineární operátor L_{TV} aplikovaný na matici spočítá její řádkové difference. Máme-li například matici $\Psi \in \mathbb{C}^{M \times N}$, matice $L_{\text{TV}} \Psi$ bude po řádcích obsahovat $\nabla \psi_{m,:}$ pro $m = 1, \dots, M$. Rozměry

matice $L_{TV}\Psi$ tedy budou $M \times N - 1$. Operátor L_{TV}^* představuje adjungovaný operátor L_{TV} . Výsledkem operace $L_{TV}^*\Psi$ by tedy byla matice o rozměrech $M \times N + 1$.

Funkce clip_1 je definována následovně

$$\text{clip}_1(x) = \frac{x}{\max\{|x|, 1\}} \quad (5.15)$$

a $\tilde{\mathbf{C}}$ v algoritmu 5.1. představuje komplexní sdružení po jednotlivých prvcích $\hat{\mathbf{C}}$.

Článek [19] říká, že konvergence algoritmu 5.1. je dosaženo, pokud

$$\tau \left(\frac{\beta}{2} + \sigma \|L^*L\| \right) < 1, \quad (5.16)$$

kde $\beta = \max |\hat{c}(m, n)|^2$. Dále si můžeme vyjádřit, že

$$\begin{aligned} \|L^*L\| &= \|\lambda_{TV}^2 L_{TV}^* L_{TV} + \lambda_1^2 \mathbf{I}\| \leq \\ &\leq \|\lambda_{TV}^2 L_{TV}^* L_{TV}\| + \|\lambda_1^2 \mathbf{I}\| = \\ &= \|\lambda_{TV} L_{TV}\|^2 + \|\lambda_1 \mathbf{I}\|^2 = \\ &= \lambda_{TV}^2 \|L_{TV}\|^2 + \lambda_1^2 \cdot 1 \leq \\ &\leq 4\lambda_{TV}^2 + \lambda_1^2, \end{aligned} \quad (5.17)$$

při $\|L_{TV}\| \leq 2$. S využitím vzorců (5.16) a (5.17) tedy získáme podmínku

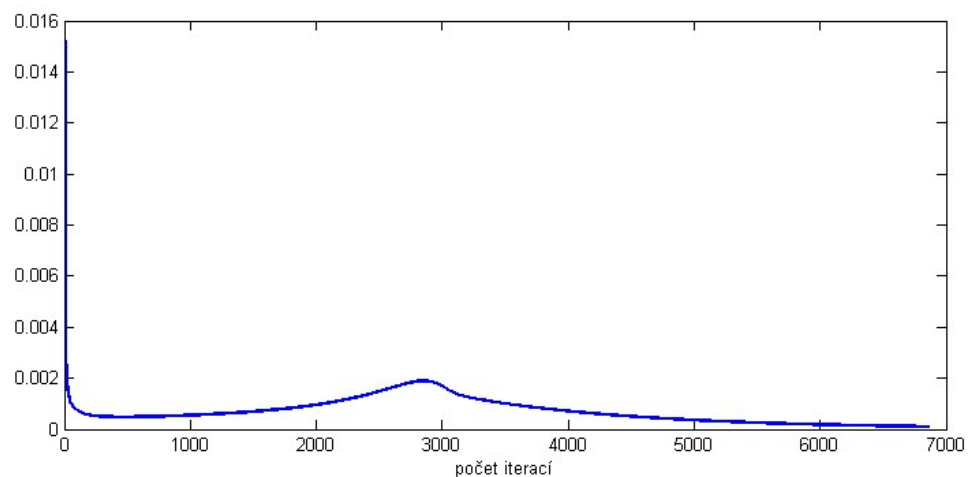
$$\sigma(4\lambda_{TV}^2 + \lambda_1^2) < \frac{1}{\tau} - \frac{\beta}{2}, \quad (5.18)$$

která nám po nastavení $\tau, \sigma > 0$ zajistí konvergenci algoritmu 5.1.

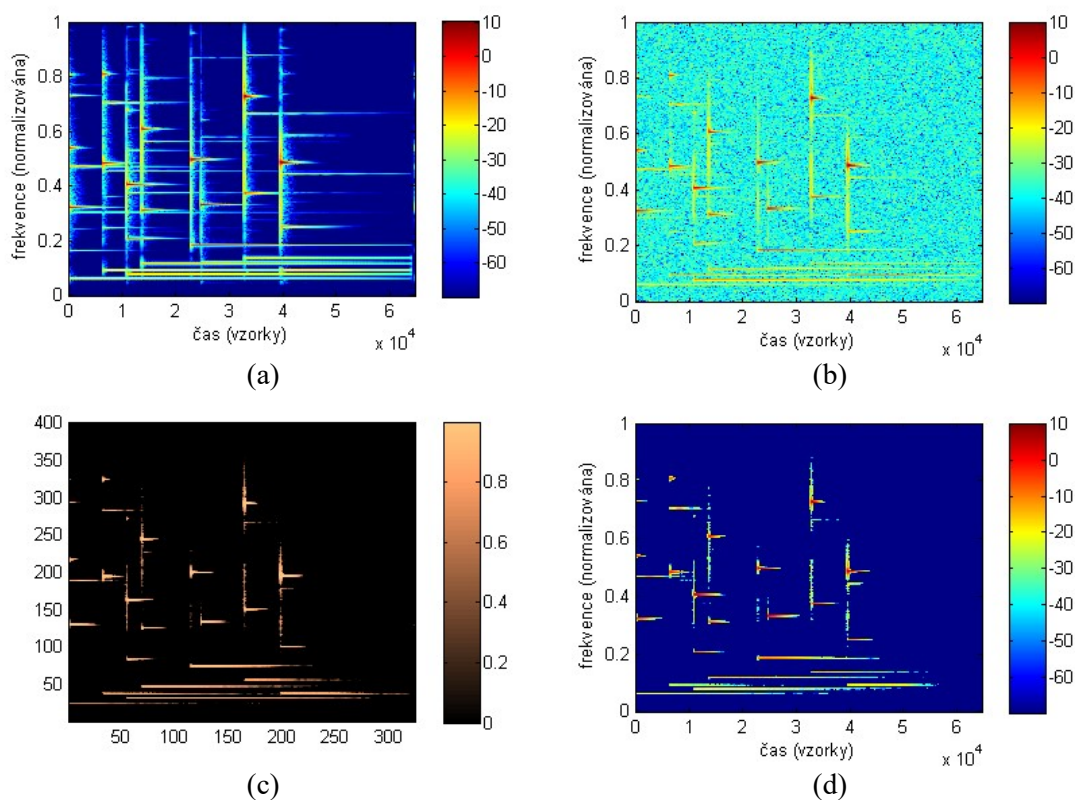
Jako kritérium ukončení iterací algoritmu jsme si zvolili velikost relativní změny $x^{(i+1)}$ v průběhu iterací. Je-li tedy změna $x^{(i+1)}$ menší než zadaná tolerance μ , iterace se ukončí. Podmínka ukončení iterací vypadá takto

$$\frac{\|x^{(i)} - x^{(i+1)}\|_F}{\|x^{(i)}\|_F} < \mu. \quad (5.19)$$

Průběh velikosti relativní změny $x^{(i+1)}$ během iterací můžeme vidět na obrázku 5.3.



Obr. 5.3: Zobrazení průběhu velikosti relativní změny $x^{(i+1)}$ během iterací algoritmu.



Obr. 5.4: Zobrazení (a) spektrogramu originálního vstupního signálu, (b) spektrogramu vstupního signálu po přidání šumu, (c) modulu řídké mapy koeficientů $\hat{a}(m, n)$ (výstup algoritmu 5.1.) a (d) časově-frekvenčních koeficientů $z(m, n) = \hat{c}(m, n)\hat{a}(m, n)$ výstupního signálu.

5.3 Moderní metoda rozšířená o druhou derivaci a vertikální totální variaci

Novým nápadem, se kterým přichází článek [19], je pokusit se při rekonstrukci využít druhou derivaci fáze. Toho dosáhneme úpravou $\alpha(m, n)$ ve vzorci (5.3). Stále platí $|\alpha(m, n)| \leq 1$, avšak pro fázi $\alpha(m, n)$ by v tomto případě platilo

$$\arg(\alpha(m, n)) = 2\arg(c(m, n)) - \arg(c(m, n+1)) - \arg(c(m, n-1)). \quad (5.20)$$

Jelikož došlo ke změně $\alpha(m, n)$ je také třeba upravit výpočet $\hat{c}(m, n)$, aby stále platilo (5.3). Upravené $\hat{c}(m, n)$ vypadá následovně

$$\begin{aligned} \hat{c}(m, n) &= \overline{c(m, n)} \frac{c(m, n-1)}{|c(m, n-1)|} \frac{c(m, n+1)}{|c(m, n+1)|} = \\ &= \overline{c(m, n)} e^{j\arg(c(m, n-1))} e^{j\arg(c(m, n+1))}. \end{aligned} \quad (5.21)$$

Na obrázku 5.5. vidíme zobrazení fáze mapy $\alpha(m, n)$ v případě užití první derivace, jako tomu je v části 5.2 a také v případě užití druhé derivace. Vidíme, že druhá derivace generuje konstantní hodnoty nejen pro horizontální směr, ale také pro vertikální směr. To nás přivádí k myšlence pokusit se toho využít a vyžadovat během minimalizace nízkou i totální variaci ve vertikálním směru. Rozšířená úloha (5.8) vypadá takto

$$\hat{\mathbf{A}} = \arg \min_{|\mathbf{A}| \leq 1} \frac{1}{2} \|\mathbf{C} - \hat{\mathbf{C}}\mathbf{A}\|_F^2 + \text{TV}(\mathbf{A}) + \lambda_1 \|\mathbf{A}\|_1. \quad (5.22)$$

Funkce TV nyní představuje dvourozměrnou izotropní totální variaci, kterou můžeme zapsat jako

$$\text{TV}(\mathbf{A}) = \|L_{\text{TV}}\mathbf{A}\|_{12} = \left\| \begin{bmatrix} \lambda_{\text{TVh}} \nabla_h \\ \lambda_{\text{TVv}} \nabla_v \end{bmatrix} \mathbf{A} \right\|_{12}, \quad (5.23)$$

kde ∇_h představuje difference \mathbf{A} po řádcích (ve směru časové osy) a ∇_v představuje difference \mathbf{A} po sloupcích (ve směru frekvenčního rozlišení). Konstanty λ_{TVh} a λ_{TVv} opět určují váhu, jaká je přikládána tomuto regularizačnímu členu. Funkce $\|\cdot\|_{12}$ představuje smíšenou maticovou normu, kdy pro matice \mathbf{U}, \mathbf{V} platí, že

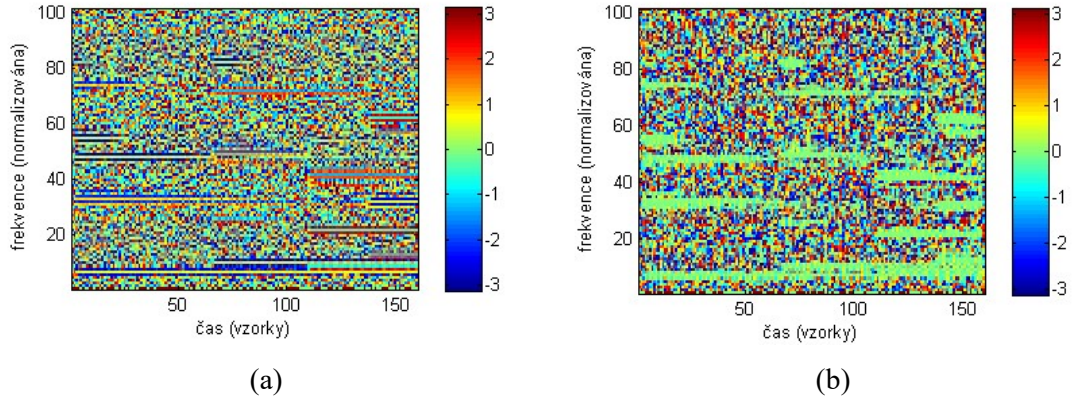
$$\left\| \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} \right\|_{12} = \left\| \sqrt{|\mathbf{U}|^2 + |\mathbf{V}|^2} \right\|_1 \quad (5.24)$$

se všemi operacemi probíhajícími po jednotlivých prvcích matic.

Výsledná podoba upravené úlohy (5.8) vypadá takto

$$\hat{\mathbf{A}} = \arg \min_{|\mathbf{A}| \leq 1} \frac{1}{2} \|\mathbf{C} - \hat{\mathbf{C}}\mathbf{A}\|_F^2 + \left\| \begin{bmatrix} \lambda_{\text{TVh}} \nabla_h \\ \lambda_{\text{TVv}} \nabla_v \end{bmatrix} \mathbf{A} \right\|_{12} + \lambda_1 \|\mathbf{A}\|_1 + \iota_{\{\mathbf{Z} \|\mathbf{Z}\|_\infty \leq 1\}}(\mathbf{A}). \quad (5.25)$$

Předpoklad je takový, že pro tonální stacionární signály budeme vynucovat nízkou převážně horizontální totální variaci (hodnota λ_{TVh} bude větší než λ_{TVv}) a pro tranzientní zvuky to bude naopak převážně vertikální totální variace (hodnota λ_{TVv} bude větší než λ_{TVh}). Stále platí, že nastavení λ_{TVh} , λ_{TVv} a λ_1 určuje míru, s jakou upřednostňujeme tyto regularizační členy před požadavkem $\|\mathbf{C} - \hat{\mathbf{C}}\mathbf{A}\|_F^2/2$, který nám zajišťuje, aby se nové časově-frekvenční koeficienty příliš nelišily od původních.



Obr. 5.5: První derivace fáze $\alpha(m, n)$ (a) a druhá derivace fáze $\alpha(m, n)$ (b).

Vidíme, že úloha (5.25) se příliš neliší od úlohy (5.12) a lze na ni opět aplikovat proximální Condatův algoritmus, kdy bude $f = \|\mathbf{C} - \hat{\mathbf{C}} \odot \cdot\|_F^2/2$, $g = \iota_{\{z \mid \|z\|_\infty \leq 1\}}$, $h_1 = \|\cdot\|_1$, $L_1 = \lambda_1 \mathbf{I}$, $h_{TV} = \|\cdot\|_{12}$ a $L_{TV} = \begin{bmatrix} \lambda_{TVh} \nabla_h \\ \lambda_{TVv} \nabla_v \end{bmatrix}$.

Konkrétní podoba Condatova algoritmu řešící (5.25) je tedy:

Algoritmus 5.2.

Vstupem jsou \mathbf{C} , $\hat{\mathbf{C}}$, λ_1 , λ_{TV} , λ_{TVh} a lineární operátor L_{TV} .

Výstupem bude $x^{(i+1)}$.

Nastaví se parametry $\tau, \sigma > 0$;

nastaví se proměnná $x^{(0)}$ a duální proměnné $u_{TV}^{(0)} = \begin{bmatrix} u_{TVh}^{(0)} \\ u_{TVv}^{(0)} \end{bmatrix}$, $u_1^{(0)}$;

vytvoří se čítač iterací $i = 0$.

V každé iteraci provede algoritmus výpočty:

$$x^{(i+1)} = \text{clip}_1 \left(x^{(i)} - \tau \left[|\hat{\mathbf{C}}|^2 \odot x^{(i)} - \bar{\mathbf{C}} \odot \mathbf{C} + \lambda_{TVh} \nabla_h^* u_{TVh}^{(i)} + \lambda_{TVv} \nabla_v^* u_{TVv}^{(i)} + \lambda_1 u_1^{(i)} \right] \right);$$

$$p = 2x^{(i+1)} - x^{(i)};$$

$$q_{TV} = u_{TV}^{(i)} + \sigma L_{TV} p;$$

$$q_1 = u_1^{(i)} + \sigma \lambda_1 p;$$

$$u_{TV}^{(i+1)} = \text{clip}_1^{\text{group}}(q_{TV});$$

$$u_1^{(i+1)} = \text{clip}_1(q_1);$$

$$i = i + 1.$$

Algoritmus každou iterací konverguje k jeho řešení. Iterace ukončíme, je-li splněno nějaké vhodné kritérium.

Výstup $x^{(i+1)}$ algoritmu 5.2. opět poté odpovídá hledanému $\hat{a}(m, n)$, které použijeme pro získání časově-frekvenční reprezentace $z(m, n) = \hat{c}(m, n)\hat{a}(m, n)$ odšuměného signálu.

Funkce clip_1 a $\tilde{\mathbf{C}}$ jsou definovány stejně jako v předchozím případě. Funkce $\text{clip}_1^{\text{group}}$ je definována následovně

$$\text{clip}_1^{\text{group}}([\mathbf{U}; \mathbf{V}]) = \frac{[\mathbf{U}; \mathbf{V}]}{\max\{\sqrt{|\mathbf{U}|^2 + |\mathbf{V}|^2}, 1\}}. \quad (5.26)$$

Konvergence algoritmu 5.2. je dosaženo, pokud

$$\tau \left(\frac{\beta}{2} + \sigma \|L_{\text{TV}}^* L_{\text{TV}} + L_1^* L_1\| \right) < 1, \quad (5.27)$$

kde opět $\beta = \max|\hat{c}(m, n)|^2$. Dále si můžeme vyjádřit, že

$$\begin{aligned} \|L_{\text{TV}}^* L_{\text{TV}} + L_1^* L_1\| &\leq \|L_{\text{TV}}^* L_{\text{TV}}\| + \|L_1^* L_1\| = \\ &= \|\lambda_{\text{TVh}}^2 \nabla_{\text{h}}^* \nabla_{\text{h}} + \lambda_{\text{TVv}}^2 \nabla_{\text{v}}^* \nabla_{\text{v}}\| + \|\lambda_1^2 \mathbf{I}\| \leq \\ &\leq \lambda_{\text{TVh}}^2 \|\nabla_{\text{h}}^* \nabla_{\text{h}}\| + \lambda_{\text{TVv}}^2 \|\nabla_{\text{v}}^* \nabla_{\text{v}}\| + \lambda_1^2 = \\ &= \lambda_{\text{TVh}}^2 \|\nabla_{\text{h}}\|^2 + \lambda_{\text{TVv}}^2 \|\nabla_{\text{v}}\|^2 + \lambda_1^2 \leq \\ &\leq 4(\lambda_{\text{TVh}}^2 + \lambda_{\text{TVv}}^2) + \lambda_1^2 \end{aligned} \quad (5.28)$$

S využitím vzorců (5.27) a (5.28) tedy získáme podmínku

$$\sigma [4(\lambda_{\text{TVh}}^2 + \lambda_{\text{TVv}}^2) + \lambda_1^2] < \frac{1}{\tau} - \frac{\beta}{2}, \quad (5.29)$$

která nám po nastavení $\tau, \sigma > 0$ zajistí konvergenci algoritmu 5.2.

Iterace algoritmu opět ukončíme, je-li relativní změna $x^{(i+1)}$ menší než zadaná tolerance μ .

6 REALIZACE ALGORITMŮ V PROSTŘEDÍ MATLAB

V této kapitole se zaměříme na realizaci navržených algoritmů pro odstranění šumu. Algoritmy budou implementovány v prostředí MATLAB s využitím toolboxu LTFAT (The Large Time-Frequency Analysis Toolbox). Jedná o toolbox vyvíjený ve spolupráci univerzit CAHR (Technical University of Denmark), ARI (Austrian Academy of Sciences) a LATP (Universite de Provence). Toolbox slouží pro práci s časově-frekvenční analýzou a syntézou signálů. Poskytuje velké množství lineárních transformací včetně Gaborovy a waveletové transformace spolu s konstrukcí okénkových funkcí a manipulací s koeficienty objektů daných transformací [20]. Důležité pro nás je především to, že jsme tak schopni vytvořit Gaborův frame. Toolbox umožňuje vytvořit matici Gaborových koeficientů. Z hlediska výpočetní náročnosti je však práce s maticemi velmi nevýhodná. Výpočty s maticemi si žádají poměrně velký výpočetní výkon a tedy obvykle i poměrně dlouhý výpočetní čas. Proto je pro nás výhodnější nepracovat s framem v podobě matice, ale v podobě objektu dané transformace. K analýze a syntéze signálu potom využíváme funkce `frana.m` a `frsyn.m`, které nám LTFAT nabízí. Tyto funkce nepočítají s maticemi, ale k výpočtům používají FFT (Rychlou Fourierovu transformaci), proto je výpočet rychlejší.

Gaborův frame vytvoříme příkazem:

```
F = frame('dgtreal', {w, WLen}, a, M);
```

Objekt `F` je objektem diskrétní Gaborovy transformace pro reálné signály. Okénkovou funkcí bude v našem kódu vždy Hannovo okno `w = 'hann'` s délkou zadanou parametrem `WLen`. Parametr `a` udává velikost posunutí okna, `M` udává počet kanálů. Pro naši realizaci je důležité, aby byl vytvořený frame tzv. 1-těsný, toho docílíme použitím příkazu

```
F = frametight(F);
```

který vrací těsný frame vytvořený z framu zadaného v jeho parametru. Byl-li frame těsný už předtím, provede pouze jeho normalizaci. Pomocí následujícího příkazu si zobrazíme hranice námi vytvořeného framu:

```
[A,B] = framebounds(F);
```

Mělo by platit $A = B = 1$, čímž si potvrdíme, že se jedná o 1-těsný frame.

Metoda odšumování využívající informace uložené v magnitudě spektrogramu je naprogramována ve funkci `odsumMag.m`, kterou můžeme vidět ve výpisu 6.1. Vstupy této funkce jsou: vstupní signál `f_in`, Gaborův frame `F`, hodnota `snr_in` a parametr `eta`. Hodnota `snr_in` slouží k přidání bílého Gaussova šumu ke vstupnímu signálu se zadaným odstupem signálu od šumu. Více se SNR (odstupu signálu od šumu) budeme věnovat v kapitole 7. V kódu se pro útlum malých magnitud používá nelineární funkce $h(x) = x/(x + \eta)$. Vstupní parametr `eta` představuje míru útlumu η . Výstupními parametry funkce `odsumMag.m` jsou: výstupní signál `f_out`, výsledný odstup signálu od šumu `snr_out` a struktura `detail`. Struktura `detail` obsahuje: vstupní signál s přidáním šumu `detail.f_in_sum`, časově-frekvenční koeficienty zašuměného signálu `detail.c_in`, časově-frekvenční koeficienty signálu po zpracování touto metodou `detail.c_out` a čas v minutách `detail.cas` potřebný pro výpočet.

Výpis kódu 6.1: odsumMag.m – Metoda odšumování signálu vycházející z magnitudy

```
function [ f_out, snr_out, detail ] = odsumMag(f_in, F, snr_in, eta)
%Metoda potlačování šumu využívající magnitud spektragramu

f_in_sum = awgn(f_in, snr_in, 'measured'); %přidání Gaussova bílého
šumu s odstupem signálu od šumu daným hodnotou snr_in

c_in = frana(F,f_in_sum); %Gaborova analýza

mag = abs(c_in); %výpočet matice magnitud
phase = angle(c_in); %výpočet fáze
a = mag./(mag+eta); %výpočet korekčních faktorů
mag = mag.*a; %násobení magnitud jednotlivými korekčními faktory
c_out = mag.*exp(1j*phase); %připojení fáze zpět k magnitudám

f_out = frsyn(F,c_out); %syntéza signálu z nových koeficientů

snr_out = 20*log10(norm(f_in)/norm(f_out - f_in)); %výpočet SNR
zrekonstruovaného signálu

detail.c_in = c_in; %časově-frekvenční koeficienty zašuměného
signálu
detail.c_out = c_out; %časově-frekvenční koeficienty výstupního
signálu
detail.f_in_sum = f_in_sum; %vstupní signál s přidaným šumem
detail.cas = toc/60; %čas potřebný pro výpočet v minutách
end
```

Moderní metoda odstranění šumu založená na fázi a využívající řídkosti je naprogramována ve funkci odsumFaz1D.m. Jedná se o metodu popsanou v části 5.2, která využívá první derivaci fáze a řádkovou totální variaci. Kód této metody implementovaný v MATLABU můžeme vidět ve výpisu 6.2. Vstupními parametry funkce odsumFaz1D.m jsou: vstupní signál f_{in} , Gaborův frame F , hodnota snr_{in} a parametry λ_{TV} a λ_1 . Výstupními parametry jsou: výstupní signál f_{out} , výsledný odstup signálu od šumu snr_{out} a struktura $detail$. Struktura $detail$ obsahuje: vstupní signál s přidaným šumem $detail.f_{in_sum}$, časově-frekvenční koeficienty zašuměného signálu $detail.c_{in}$, časově-frekvenční koeficienty signálu po zpracování touto metodou $detail.c_{out}$, počet iterací $detail.pocetIter$, výslednou řídkou mapu $\alpha(m,n)$ $detail.mapaAlfa$, vektor $detail.historie$ obsahující velikost relativní změny $\alpha(m,n)$ v průběhu iterací a čas v minutách $detail.cas$ potřebný pro výpočet.

Výpis kódu 6.2: odsumFaz1D.m – Metoda odšumování signálu založená na první derivaci fáze a využívající řídkosti

```
function [ f_out, snr_out, detail ] = odsumFaz1D( f_in, F, snr_in,
lambdaTV, lambda1 )
%Metoda potlačování šumu založená na první derivaci fáze a
využívající řídkosti
tic
f_in_sum = awgn(f_in, snr_in, 'measured'); %přidání Gaussova bílého
šumu s odstupem signálu od šumu snr_in
c_in = frana(F,f_in_sum); %Gaborova analýza
```

```

c = framecoef2native(F,c_in); %převedení časově-frekvenčních
koeficientů do matice
c_striska = abs(c).*exp(1j*angle(circshift(c,[0,1]))); %výpočet c se
stříškou
%výpočty nutné pro zajištění konvergence algoritmu
beta = max(max(abs(c_striska).^2));
sigma = 100000;
tau = 0.999*(2/(2*sigma*(4*lambdaTV^2 + lambda1^2)+beta));
%inicializace primární proměnné a duálních proměnných
x = ones(size(c));
uTV = ones(size(x,1),size(x,2)-1);
u1 = ones(size(x));
i = 1; %čítač iterací
maxIter = 15000; %nastavení maximálního počtu iterací
tolerance = 1e-4; %nastavení tolerance pro ukončení algoritmu
historie = zeros(maxIter,1); %vektor pro uložení relativní změny x v
průběhu iterací
clip1 = @(x) x./max(abs(x),1); %funkce clip1
diferenceRadkove = @(x) diff(x,1,2); %funkce pro výpočet řádkových
diferencí
diferenceRadkoveAdj = @(x) -diff([zeros(size(x,1),1) x
zeros(size(x,1),1)],1,2); %adjungovaný operátor výpočtu řádkových
diferencí
while(i<maxIter) %iterační Condatův algoritmus
    x1 = clip1(x - tau*((abs(c_striska)).^2.*x - conj(c_striska).*c
+ lambdaTV*diferenceRadkoveAdj(uTV) + lambda1*u1));
    p = 2*x1 - x;
    qTV = uTV + sigma*lambdaTV*diferenceRadkove(p);
    q1 = u1 + sigma*lambda1*p;
    uTV = clip1(qTV);
    u1 = clip1(q1);
    historie(i) = norm(x-x1,'fro')/norm(x,'fro'); %velikost
relativní změny x v průběhu iterací
    if i>5000 && historie(i)<tolerance
        x = x1;
        break %ukončení iterací, pokud je relativní změna x menší
než zadaná tolerance a současně se jedná alespoň o 5001. iteraci
    end
    x = x1;
    i = i+1;
end
c_out = framenative2coef(F, x.*c_striska); %přenásobení časově-
frekvenčních koeficientů řádkovou mapou x
f_out = frsyn(F,c_out); %syntéza signálu z nových koeficientů
snr_out = 20*log10(norm(f_in)/norm(f_out - f_in)); %výpočet SNR
zrekonstruovaného signálu

detail.c_in = c_in; %časově-frekvenční koeficienty zašuměného
signálu
detail.c_out = c_out; %časově-frekvenční koeficienty výstupního
signálu
detail.f_in_sum = f_in_sum; %vstupní signál s přidáním šumem

```

```

detail.pocetIter = i; %počet iterací
detail.mapaAlpha = x; %výsledná řádká mapa alpha
detail.historie = historie(1:i-1); %relativní změna alpha v průběhu
iterací
detail.cas = toc/60; %čas potřebný pro výpočet v minutách
end

```

Moderní metoda odstranění šumu založená na druhé derivaci fáze a využívající řídkosti je naprogramována ve funkci `odsumFaz2D.m`. Kód této metody implementovaný v MATLABU můžeme vidět ve výpisu 6.3. Vstupními parametry funkce `odsumFaz2D.m` jsou obdobně jako v předchozím případě: vstupní signál `f_in`, Gaborův frame `F`, hodnota `snr_in` a parametry `lambdaTVh` pro řádkovou totální variaci, `lambdaTVv` pro sloupcovou totální variaci a parametr `lambda1`. Výstupní parametry jsou stejné jako u funkce `odsumFaz1D.m`, jsou to tedy: výstupní signál `f_out`, výsledný odstup signálu od šumu `snr_out` a struktura `detail`, která obsahuje `detail.f_in_sum`, `detail.c_in`, `detail.c_out`, `detail.pocetIter`, `detail.mapaAlpha`, `detail.historie` a `detail.cas`.

Výpis kódu 6.3: `odsumFaz2D.m` – Metoda odšumování signálu založená na druhé derivaci fáze a využívající řídkosti

```

function [ f_out, snr_out, detail ] = odsumFaz2D( f_in, F, snr_in,
lambdaTVh, lambdaTVv, lambda1 )
%Metoda potlačování šumu založená na první derivaci fáze a
využívající řídkosti
tic
f_in_sum = awgn(f_in, snr_in, 'measured'); %přidání Gaussova bílého
šumu s odstupem signálu od šumu snr_in
c_in = frana(F,f_in_sum); %Gaborova analýza
c = framecoef2native(F,c_in); %převod časově-frekvenčních
koeficientů do matice
%výpočet c se stříškou
c_left = circshift(c,[0,1]); Phleft = c_left./abs(c_left);
Phleft(isnan(Phleft)) = 0;
c_right = circshift(c,[0,-1]); Phright = c_right./abs(c_right);
Phright(isnan(Phright)) = 0;
c_striska = conj(c).*Phright.*Phleft;
%výpočty nutné pro zajištění konvergence algoritmu
beta = max(max(abs(c_striska).^2));
sigma = 8000;
tau = 0.999*(2/(2*sigma*(4*(lambdaTVh^2 + lambdaTVv^2) +
lambda1^2)+beta));
%inicializace primární proměnné a duálních proměnných
x = ones(size(c));
uTVh = ones(size(x,1),size(x,2)-1);
uTVv = ones(size(x,1)-1,size(x,2));
u1 = ones(size(x));
i = 1; %čítač iterací
maxIter = 20000; %nastavení maximálního počtu iterací
tolerance = 1e-4; %nastavení tolerance pro ukončení algoritmu
historie = zeros(maxIter,1); %vektor pro uložení relativní změny x v
průběhu iterací
clip1 = @(x) x./max(abs(x),1); %funkce clip1

```

```

clip1_group = @(x) x./([sqrt(max((x(1:end/2,:).^2 +
x(end/2+1:end,:).^2),1)); sqrt(max((x(1:end/2,:).^2 +
x(end/2+1:end,:).^2),1))]); %funkce clip1_group

diferenceRadkove = @(x) diff(x,1,2); %funkce pro výpočet řádkových
diferencí
diferenceSloupcove = @(x) diff(x,1,1); %funkce pro výpočet
sloupcových diferencí
diferenceRadkoveAdj = @(x) -diff([zeros(size(x,1),1) x
zeros(size(x,1),1)],1,2); %adjungovaný operátor výpočtu řádkových
diferencí
diferenceSloupcoveAdj = @(x) -diff([zeros(1,size(x,2)); x;
zeros(1,size(x,2))],1,1); %adjungovaný operátor výpočtu sloupcových
diferencí
while(i<maxIter) %iterační Condatův algoritmus

    x1 = clip1(x - tau*((abs(c_striska)).^2.*x - conj(c_striska)).*c
+ lambdaTVh*diferenceRadkoveAdj(uTVh) +
lambdaTVv*diferenceSloupcoveAdj(uTVv) + lambda1*u1));

    p = 2*x1 - x;
    qTVh = [uTVh + sigma*lambdaTVh*diferenceRadkove(p)
zeros(size(uTVh,1),1)];
    qTVv = [uTVv + sigma*lambdaTVv*diferenceSloupcove(p); zeros(1,
size(uTVv,2))];
    q1 = u1 + sigma*lambda1*p;
    uTV = clip1_group([qTVh; qTVv]);
    uTVh = uTV(1:end/2,1:end-1);
    uTVv = uTV(end/2+1:end-1,:);
    u1 = clip1(q1);
    historie(i) = norm(x-x1,'fro')/norm(x,'fro'); %velikost
relativní změny x v průběhu iterací
    if i>5000 && historie(i)<tolerance
        x = x1;
        break %ukončení iterací, pokud je relativní změna x menší
než zadaná tolerance a současně se jedná alespoň o 5001. iteraci
    end
    x = x1;
    i = i+1;
end
c_out = framenative2coef(F, x.*c_striska); %přenásobení časově-
frekvenčních koeficientů řádkou mapou x
f_out = frsyn(F,c_out); %syntéza signálu z nových koeficientů
snr_out = 20*log10(norm(f_in)/norm(f_out - f_in)); %výpočet SNR
zrekonstruovaného signálu

detail.c_in = c_in; %časově-frekvenční koeficienty zašuměného
signálu
detail.c_out = c_out; %časově-frekvenční koeficienty výstupního
signálu
detail.f_in_sum = f_in_sum; %vstupní signál s přidáním šumem
detail.pocetIter = i; %počet iterací

```

```
detail.mapaAlpha = x; %výsledná řídká mapa alpha  
detail.historie = historie(1:i-1); %relativní změna alpha v průběhu  
iterací  
detail.cas = toc/60; %čas potřebný pro výpočet v minutách  
end
```

7 TESTOVÁNÍ NAVRŽENÝCH ALGORITMŮ

V této kapitole se zaměříme na testování výše navržených a implementovaných algoritmů. Nejdříve otestujeme každý zvlášť a později provedeme komplexnější test na více vstupních signálech a s různým nastavením jednotlivých parametrů algoritmů. Během testování budou parametry Gaborovy transformace vždy: velikost okna $W_{Len} = 800$, typ okna Hannovo, posunutí okna $a = 200$ a počet kanálů $M = 800$.

7.1 Hodnocení úspěšnosti rekonstrukce pomocí SNR

Objektivním kritériem hodnocení úspěšnosti rekonstrukce zašuměných signálů pro nás bude SNR (odstup signálu od šumu). Ten nám udává poměr výkonu užitečného signálu P_s ku výkonu signálu neužitečného P_n [21]. V našem případě je užitečným signálem originální vstupní signál bez šumu. Z toho je patrné, že tuto metodu hodnocení úspěšnosti rekonstrukce je možné použít pouze v případě, že originální signál známe. V našich experimentech tomu tak vždy bude, jelikož budeme k účelu testování přidávat k ne zašuměným datům šum vždy uměle. Neužitečným signálem je v našem případě rozdíl výstupního zrekonstruovaného signálu a originálního vstupního signálu bez šumu. Hodnota SNR se udává v decibelech, čím je hodnota větší, tím je rekonstrukce úspěšnější. Obecný vzorec pro výpočet SNR je následující

$$SNR = 10 \log \frac{P_s}{P_n} = 10 \log \frac{\sum_{m=1}^{l-1} s^2[m]}{\sum_{m=1}^{l-1} n^2[m]}, \quad (7.1)$$

kde l je délka signálu, $s[m]$ je vzorek užitečného signálu a $n[m]$ je vzorek neužitečného signálu [22]. S využitím definice 1.2. si tuto rovnici můžeme přepsat do tvaru

$$SNR = 10 \log \frac{\|\mathbf{s}\|_2^2}{\|\mathbf{n}\|_2^2} = 20 \log \frac{\|\mathbf{s}\|_2}{\|\mathbf{n}\|_2}. \quad (7.2)$$

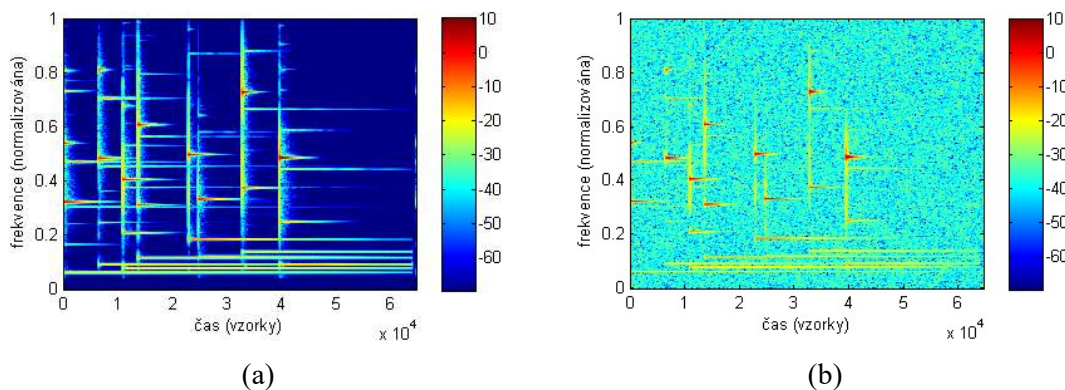
V našem případě vypadá výpočet SNR takto

$$SNR = 20 \log \frac{\|\mathbf{y}_{orig}\|_2}{\|\mathbf{y}_{výst} - \mathbf{y}_{orig}\|_2}, \quad (7.3)$$

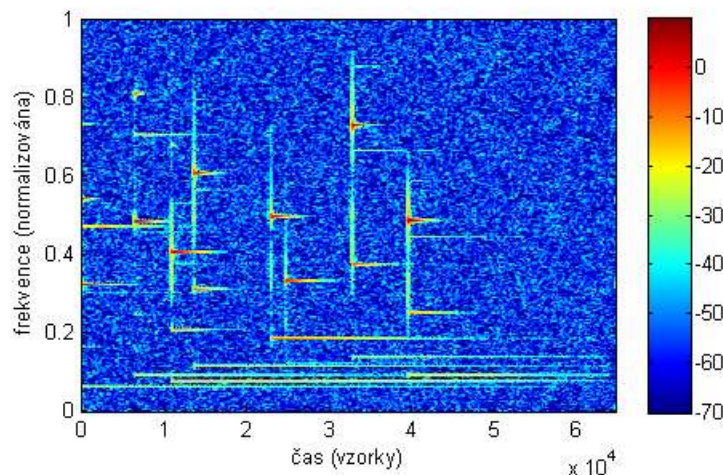
kde \mathbf{y}_{orig} představuje vstupní signál bez šumu a $\mathbf{y}_{výst}$ představuje výstupní signál po zpracování zašuměného signálu danou metodou.

7.2 Test metody využívající magnitudy spektrogramu

Následuje realizace experimentu odstranění šumu z nahrávky zvonkohry za pomoci metody využívající informací uložených v magnitudě spektrogramu, kterou jsme si představili v části 5.1. Ke vstupnímu signálu zvonkohry byl přidán bílý Gaussův šum s odstupem signálu od šumu 10 dB. Spektrogram originálního signálu bez šumu vidíme na obrázku 7.1 (a). Spektrogram zašuměného vstupního signálu vidíme na obrázku 7.1 (b). K potlačení šumu použijeme nelineární funkci $h(x) = x/(x + \eta)$ s parametrem $\eta = 0,1$. Spektrogram výstupního signálu vidíme na obrázku 7.2.



Obr. 7.1: Spektrogram originálního vstupního signálu bez šumu (a) a spektrogram zašuměného vstupního signálu (b).

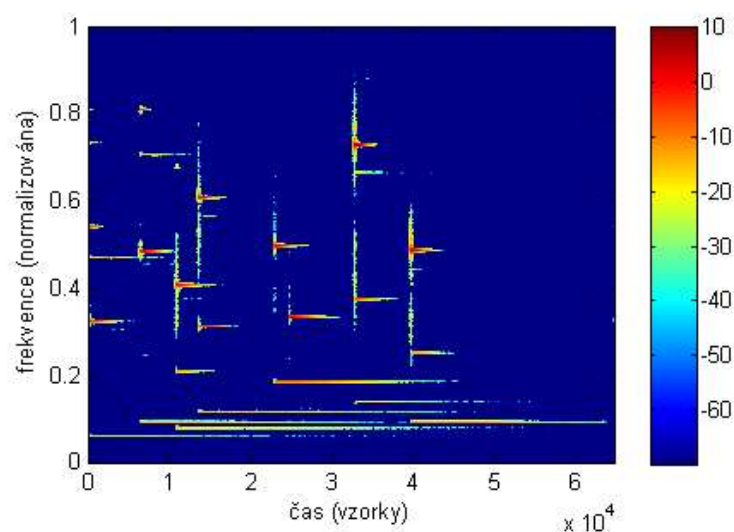


Obr. 7.2: Spektrogram výstupního signálu po zpracování zašuměných dat metodou odstranění šumu využívající magnitud spektrogramu.

Při poslechu zašuměného signálu, je přidáný šum nesporně výrazně slyšet a působí rušivě. Poslechneme-li si signál po zpracování touto metodou, je znát, že šumu je méně a působí tedy i méně rušivě. Přesto je však v nahrávce stále bez problému rozeznatelný. Navíc je při porovnání původního nezašuměného signálu a signálu na výstupu této metody slyšet mírná změna barvy původního signálu. Jedná se především o úbytek vyšších frekvencí. Výsledný odstup signálu od šumu je 12,4 dB. Doba výpočtu byla 0,002 minuty.

7.3 Test metody založené na první derivaci fáze a využívající řídkosti

Nyní aplikujeme na stejný vstupní signál nahrávky zvonkohry moderní metodu potlačení šumu založenou na první derivaci fáze a využívající řídkosti, kterou jsme si představili v části 5.2. Vstupní signál bude opět zarušen bílým Gaussovým šumem s odstupem signálu od šumu 10 dB. Parametry λ_{TV} a λ_1 budou nastaveny takto: $\lambda_{TV} = 0,0001$ a $\lambda_1 = 0,003$. Spektrogram originálního vstupního signálu bez šumu a spektrogram zašuměného vstupního signálu vidíme na



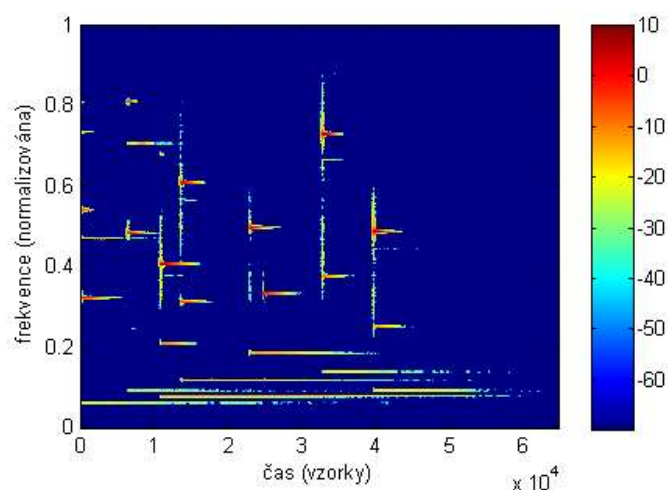
Obr. 7.3: Spektrogram výstupního signálu po zpracování zašuměných dat metodou odstranění šumu založenou na první derivaci fáze a využívající řídkosti.

obrázku 7.1. Na obrázku 7.3 můžeme vidět spektrogram výstupního signálu po zpracování touto metodou. Je zde vidět, že šum byl zcela odstraněn. Patrné je však i to, že došlo již k výrazné změně barvy zvuku zvonkohry. Při poslechu však tato změna není až tak velká a osobně mi přijde méně rušivá než původní šum, který v nahrávce byl. Výsledné SNR je 14,6 dB. Počet iterací algoritmu byl v tomto případě 5001 a doba výpočtu byla 2,2 minuty.

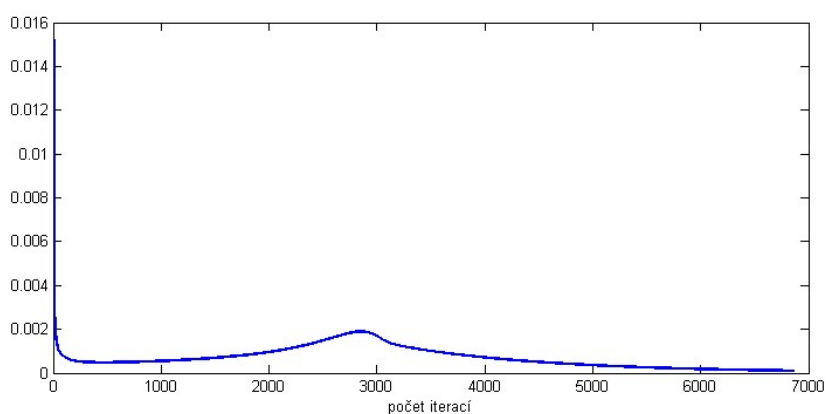
7.4 Test metody založené na druhé derivaci fáze a využívající řídkosti

Opět použijeme stejný vstupní signál zvonkohry, na který aplikujeme moderní metodu odstranění šumu založenou na druhé derivaci fáze a využívající řídkosti. Vstupní signál bude opět zarušen bílým Gaussovým šumem s odstupem signálu od šumu 10 dB. Parametry λ_{TVh} , λ_{TVv} a λ_1 budou nastaveny takto: $\lambda_{TVh} = 0,0001$, $\lambda_{TVv} = 0,0002$ a $\lambda_1 = 0,003$. Spektrogram originálního vstupního signálu bez šumu a spektrogram zašuměného vstupního signálu vidíme na obrázku 7.1. Na obrázku 7.4. můžeme vidět spektrogram výstupního signálu po zpracování touto metodou. Opět můžeme vidět, že šum byl zcela odstraněn, avšak taktéž došlo k výrazné změně barvy zvuku zvonkohry. Výsledek je velmi podobný výstupu předchozí metody. Výsledné SNR je 14,4 dB. Počet iterací algoritmu byl v tomto případě opět 5001 a doba výpočtu byla 6,3 minuty.

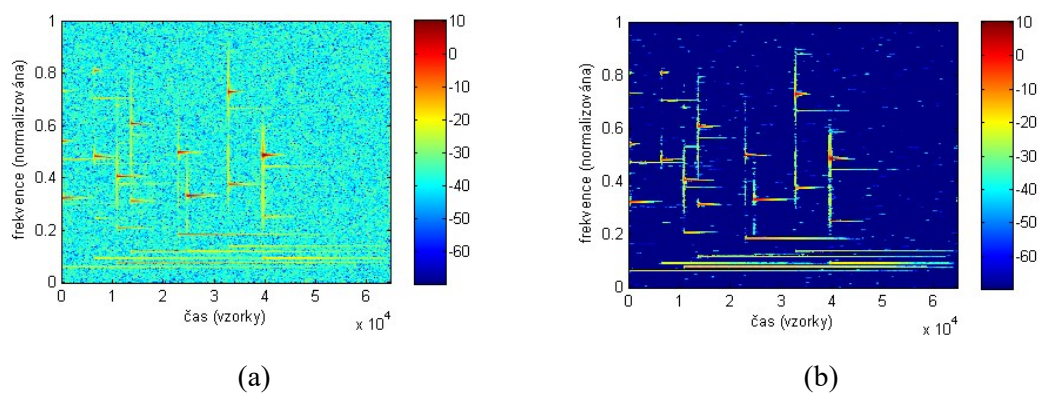
Během testování obou metod využívajících Condatův proximální algoritmus jsem si všiml, že podmínka pro ukončení iterací byla mnohdy splněna dříve, než daný problém skutečně v nějaké rozumné míře zkonvergoval. Tato skutečnost je zřejmá z obrázku 7.5, na kterém je vykreslena relativní změna mapy $\alpha(m, n)$ v průběhu iterací. Z obrázku je patrné, že tato relativní změna může být menší než zadaná tolerance i po malém počtu iterací. Zobrazíme-li si však v tuto chvíli výsledek, zjistíme, že je velmi neuspokojivý, viz obrázek 7.6 (a). Necháme-li však algoritmus iterovat dále, vidíme, že se relativní změna $\alpha(m, n)$ v průběhu iterací začne znovu zvětšovat a poté zase zmenšovat. Jako uspokojivý výsledek považuji až ten, kdy je zadaná podmínka splněna znovu. Takovýto výsledek můžeme vidět na obrázku 7.6 (b). K tomuto nárůstu a následnému poklesu relativní změny $\alpha(m, n)$ obvykle dochází mezi prvními 5000 iteracemi, proto byla k podmínce, že musí být tato relativní změna menší než zadaná tolerance přidána další podmínka a to taková, že se pro ukončení iterací musí jedna alespoň o 5001. iteraci.



Obr. 7.4: Spektrogram výstupního signálu po zpracování zašuměných dat metodou odstranění šumu založenou na druhé derivaci fáze a využívající řidkosti.



Obr. 7.5: Zobrazení průběhu velikosti relativní změny $\alpha(m, n)$ během iterací algoritmu.



Obr. 7.6: Spektrogram výstupního signálu po 400 iteracích (a) a spektrogram výstupního signálu po 5060 iteracích (b).

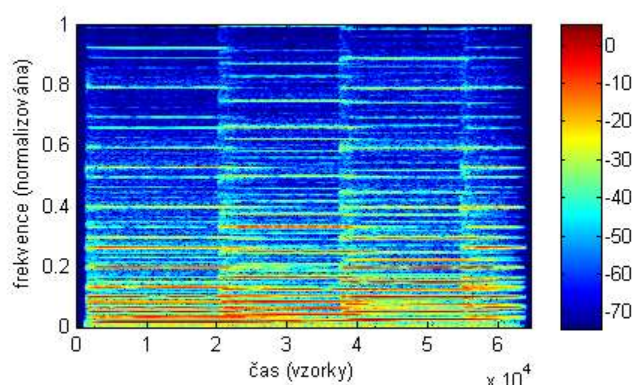
7.5 Rozšířené testování rekonstrukčních metod

V této části aplikujeme všechny tři metody odstraňování šumu, které byli v práci popsány, na tři různé vstupní signály s různým nastavením parametrů daných metod a s dvěma různými úrovněmi nežádoucího šumu. Vstupní signály budou zarušeny bílým Gaussovým šumem s odstupem signálu od šumu 15 dB a 10 dB. Úspěšnost rekonstrukce budeme hodnotit pomocí SNR společně s krátkým subjektivním hodnocením.

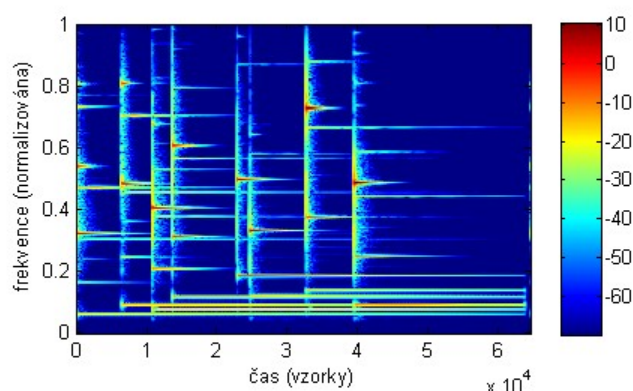
První ze vstupních signálů, které jsme vybrali je nahrávka varhan. Tato nahrávka byla vybrána proto, protože obsahuje hodně stacionárních harmonických komponent a málo tranzientních zvuků. Předpokladem tedy je, že metoda odstranění šumu založená na první derivaci fáze bude vykazovat dobré výsledky při aplikování na zašuměná data této vstupní nahrávky. Spektrogram nahrávky varhan můžeme vidět na obrázku 7.7.

Druhým z vybraných vstupních signálů je nahrávka zvonkohry. Její spektrogram můžeme vidět na obrázku 7.8. V porovnání se spektrogramem varhan vidíme, že zvonkohra obsahuje také poměrně dost stacionárních harmonických komponent, avšak také hodně tranzientních komponent, především při úderu paličky na kámen zvonkohry. Předpokládáme tedy, že pro zašuměná data tohoto vstupního signálu získáme lepší výsledky pro metodu odstranění šumu založenou na druhé derivaci, která během výpočtu využívá i vertikální totální variaci.

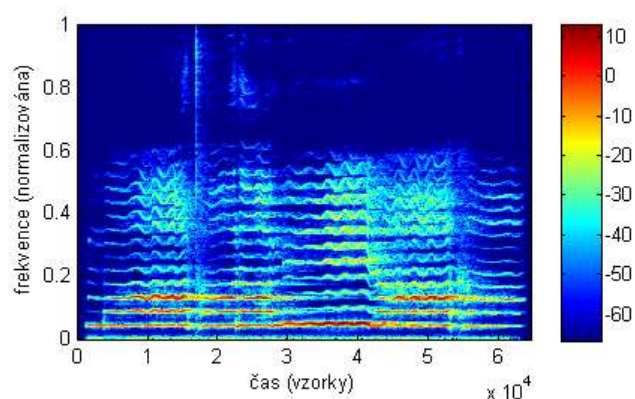
Třetím vybraným vstupním signálem je nahrávka ženského zpěvu. Jak můžeme vidět na obrázku 7.9, ve spektrogramu této nahrávky není tolik stacionárních harmonických komponent a signál je celkově oproti předchozím složitější a v průběhu času se poměrně výrazně mění.



Obr. 7.7: Spektrogram nahrávky varhan.



Obr. 7.8: Spektrogram nahrávky zvonkohry.



Obr. 7.9: Spektrogram nahrávky ženského zpěvu.

V následujících tabulkách vidíme výsledky tří metod odstranění šumu aplikovaných na vstupní signál nahrávky varhan.

Metoda využívající informací uložených v magnitudě					
$SNR_{\text{vstup}} = 15 \text{ dB}$			$SNR_{\text{vstup}} = 10 \text{ dB}$		
η	$SNR_{\text{výstup}}$	dobu výpočtu (minuty)	η	$SNR_{\text{výstup}}$	dobu výpočtu (minuty)
0,03	18,62	0,0018	0,05	14,23	0,0018
0,05	17,83	0,0019	0,1	13,61	0,0017
0,1	14,81	0,0019	0,2	10,94	0,0019

Metoda založená na první derivaci fáze							
$SNR_{\text{vstup}} = 15 \text{ dB}$				$SNR_{\text{vstup}} = 10 \text{ dB}$			
λ_1	λ_{TV}	$SNR_{\text{výstup}}$	dobu výpočtu (minuty)	λ_1	λ_{TV}	$SNR_{\text{výstup}}$	dobu výpočtu (minuty)
0,0005	0,001	23,12	2,91	0,005	0,001	17,89	2,24
0,0005	0,003	23,01	6,13	0,005	0,003	17,71	2,22
0,0005	0,005	18,79	2,22	0,005	0,005	17,68	2,22
0,002	0,001	21,00	2,20	0,01	0,001	16,09	2,25
0,002	0,003	20,69	2,23	0,01	0,003	15,96	2,20
0,002	0,005	20,72	2,99	0,01	0,005	15,86	2,20
0,005	0,001	18,56	2,21	0,03	0,001	12,17	2,21
0,005	0,003	18,47	2,20	0,03	0,003	12,10	2,20
0,005	0,005	18,37	2,21	0,03	0,005	12,18	2,21

Metoda založená na druhé derivaci fáze									
$SNR_{vstup} = 15 \text{ dB}$					$SNR_{vstup} = 10 \text{ dB}$				
λ_1	λ_{TVh}	λ_{TVv}	$SNR_{výstup}$	dobu výpočtu (minuty)	λ_1	λ_{TVh}	λ_{TVv}	$SNR_{výstup}$	dobu výpočtu (minuty)
0,0005	0,001	0,001	21,91	26,09	0,005	0,001	0,001	17,45	24,90
0,0005	0,001	0,002	20,97	25,83	0,005	0,001	0,002	17,04	24,82
0,0005	0,002	0,001	21,99	25,65	0,005	0,002	0,001	17,49	25,02
0,0005	0,002	0,002	21,14	25,50	0,005	0,002	0,002	17,18	24,93
0,002	0,001	0,001	20,21	25,15	0,01	0,001	0,001	15,79	6,22
0,002	0,001	0,002	19,45	25,07	0,01	0,001	0,002	15,44	24,79
0,002	0,002	0,001	20,16	25,07	0,01	0,002	0,001	15,74	24,66
0,002	0,002	0,002	19,46	25,08	0,01	0,002	0,002	15,52	24,81
0,005	0,001	0,001	18,14	6,47	0,03	0,001	0,001	11,72	6,12
0,005	0,001	0,002	17,70	24,69	0,03	0,001	0,002	11,75	6,18
0,005	0,002	0,001	18,10	24,68	0,03	0,002	0,001	11,75	6,19
0,005	0,002	0,002	17,62	25,00	0,03	0,002	0,002	11,62	6,18

U každé metody je zeleně označen výsledek s nejlepším SNR. Modře je pak označen výsledek, který dle mého subjektivního názoru zní při poslechu nejlépe. U všech tří metod dochází k tomu, že pro výsledky s nejlepším SNR není šum zcela odstraněn, ba ani natolik potlačen, aby již nepůsobil rušivě. To je důvodem, proč se mé subjektivní hodnocení neshoduje s objektivním kritériem založeným na SNR. Dle mého názoru je změna barvy zvuku ve výsledné nahrávce méně rušivá, než šum na pozadí.

Následují tabulky výsledků tří metod odstranění šumu aplikovaných na vstupní signál nahrávky zvonkohry.

Metoda využívající informací uložených v magnitudě					
$SNR_{vstup} = 15 \text{ dB}$			$SNR_{vstup} = 10 \text{ dB}$		
η	$SNR_{výstup}$	dobu výpočtu (minuty)	η	$SNR_{výstup}$	dobu výpočtu (minuty)
0,03	18,01	0,0018	0,05	14,71	0,0018
0,05	15,81	0,0019	0,1	12,42	0,0021
0,1	12,64	0,0018	0,2	9,83	0,0018

Metoda založená na první derivaci fáze							
$SNR_{vstup} = 15 \text{ dB}$				$SNR_{vstup} = 10 \text{ dB}$			
λ_1	λ_{TV}	$SNR_{výstup}$	dobu výpočtu (minuty)	λ_1	λ_{TV}	$SNR_{výstup}$	dobu výpočtu (minuty)
0,0002	0,0001	18,40	2,30	0,001	0,0001	18,48	3,25
0,0002	0,0003	18,43	2,30	0,001	0,0003	18,22	3,00
0,0002	0,0005	18,30	2,30	0,001	0,0005	18,12	2,95
0,0004	0,0001	22,04	6,50	0,003	0,0001	14,55	2,22
0,0004	0,0003	21,49	6,46	0,003	0,0003	14,52	2,20
0,0004	0,0005	20,82	6,53	0,003	0,0005	14,34	2,22
0,0006	0,0001	20,93	4,34	0,005	0,0001	12,76	2,20
0,0006	0,0003	20,37	4,30	0,005	0,0003	12,67	2,20
0,0006	0,0005	19,90	4,35	0,005	0,0005	12,58	2,21

Metoda založená na druhé derivaci fáze									
$SNR_{vstup} = 15 \text{ dB}$					$SNR_{vstup} = 10 \text{ dB}$				
λ_1	λ_{TVh}	λ_{TVv}	$SNR_{výstup}$	dobu výpočtu (minuty)	λ_1	λ_{TVh}	λ_{TVv}	$SNR_{výstup}$	dobu výpočtu (minuty)
0,0002	0,0001	0,0001	18,74	6,42	0,001	0,0001	0,0001	18,26	10,57
0,0002	0,0001	0,0002	18,80	6,42	0,001	0,0001	0,0002	17,96	10,40
0,0002	0,0002	0,0001	18,80	6,43	0,001	0,0002	0,0001	18,06	10,33
0,0002	0,0002	0,0002	18,70	6,28	0,001	0,0002	0,0002	17,93	10,28
0,0004	0,0001	0,0001	21,48	24,37	0,003	0,0001	0,0001	14,52	6,34
0,0004	0,0001	0,0002	21,04	23,81	0,003	0,0001	0,0002	14,35	6,34
0,0004	0,0002	0,0001	21,30	22,37	0,003	0,0002	0,0001	14,34	6,33
0,0004	0,0002	0,0002	20,70	23,50	0,003	0,0002	0,0002	14,30	6,34
0,0006	0,0001	0,0001	20,32	16,63	0,005	0,0001	0,0001	12,68	6,23
0,0006	0,0001	0,0002	19,99	16,48	0,005	0,0001	0,0002	12,61	2,20
0,0006	0,0002	0,0001	20,15	15,55	0,005	0,0002	0,0001	12,61	6,23
0,0006	0,0002	0,0002	19,71	16,00	0,005	0,0002	0,0002	12,58	6,23

Následují tabulky výsledků tří metod odstranění šumu aplikovaných na vstupní signál nahrávky ženského zpěvu.

Metoda využívající informací uložených v magnitudě					
$SNR_{vstup} = 15 \text{ dB}$			$SNR_{vstup} = 10 \text{ dB}$		
η	$SNR_{výstup}$	dobu výpočtu (minuty)	η	$SNR_{výstup}$	dobu výpočtu (minuty)
0,03	19,62	0,0018	0,05	14,94	0,0019
0,05	20,10	0,0019	0,1	15,97	0,0018
0,1	18,57	0,0019	0,2	14,65	0,0018

Metoda založená na první derivaci fáze							
$SNR_{vstup} = 15 \text{ dB}$				$SNR_{vstup} = 10 \text{ dB}$			
λ_1	λ_{TV}	$SNR_{výstup}$	dobu výpočtu (minuty)	λ_1	λ_{TV}	$SNR_{výstup}$	dobu výpočtu (minuty)
0,0005	0,001	19,12	2,23	0,005	0,001	20,13	2,25
0,0005	0,003	18,60	2,21	0,005	0,003	19,78	2,23
0,0005	0,005	18,09	2,22	0,005	0,005	19,51	2,75
0,002	0,001	22,55	3,18	0,01	0,001	19,00	2,23
0,002	0,003	21,75	3,77	0,01	0,003	18,70	2,21
0,002	0,005	21,19	5,21	0,01	0,005	18,46	2,21
0,005	0,001	20,69	2,21	0,03	0,001	16,71	2,21
0,005	0,003	20,16	2,24	0,03	0,003	16,63	2,20
0,005	0,005	19,94	2,39	0,03	0,005	16,55	2,23

Metoda založená na druhé derivaci fáze									
$SNR_{vstup} = 15 \text{ dB}$					$SNR_{vstup} = 10 \text{ dB}$				
λ_1	λ_{TVh}	λ_{TVv}	$SNR_{výstup}$	dobu výpočtu (minuty)	λ_1	λ_{TVh}	λ_{TVv}	$SNR_{výstup}$	dobu výpočtu (minuty)
0,0005	0,001	0,001	22,45	27,61	0,005	0,001	0,001	20,02	6,66
0,0005	0,001	0,002	22,45	26,44	0,005	0,001	0,002	19,73	11,76
0,0005	0,002	0,001	22,87	26,38	0,005	0,002	0,001	19,82	25,32
0,0005	0,002	0,002	22,67	25,71	0,005	0,002	0,002	19,71	25,42
0,002	0,001	0,001	22,31	9,88	0,01	0,001	0,001	18,87	6,30
0,002	0,001	0,002	21,63	25,10	0,01	0,001	0,002	18,82	6,30
0,002	0,002	0,001	21,73	25,24	0,01	0,002	0,001	18,72	6,30
0,002	0,002	0,002	21,42	25,17	0,01	0,002	0,002	18,69	6,30
0,005	0,001	0,001	20,47	6,35	0,03	0,001	0,001	16,47	6,14
0,005	0,001	0,002	20,18	6,26	0,03	0,001	0,002	16,43	6,13
0,005	0,002	0,001	20,33	6,28	0,03	0,002	0,001	16,43	6,15
0,005	0,002	0,002	20,07	6,48	0,03	0,002	0,002	16,29	6,14

Ze všech výše popsaných pozorování vidíme, že objektivní hodnocení pomocí SNR se příliš neshoduje s mým subjektivním hodnocením. Subjektivní hodnocení je však velmi závislé na jedinci a je možné, že někdo by lépe hodnotil nahrávky s pozůstatky šumu a jako rušivější faktor ve výsledku by považoval výraznou změnu barvy zvuku.

Na základě objektivního hodnocení pomocí SNR i na základě poslechu vidíme, že metoda využívající informaci uložených v magnitudě spektrogramu nedosahuje takových výsledků, jako zbylé dvě metody založené na fázi. Avšak se nám také nepodařilo prokázat, že by některá z metod založených na fázi byla pro některý z výše použitých vstupních signálů výrazně vhodnější než jiná. To platí i o předpokládaném rozdílu nahrávky varhan a zvonkohry, kdy jsme předpokládali, že pro zvonkohru bude výhodnější užití metody založené na druhé derivaci fáze. Naopak vidíme, že v porovnání s metodou využívající první derivaci fáze je tato metoda často výpočetně náročnější a přesto nedosahuje lepších výsledků.

ZÁVĚR

Cílem práce bylo, seznámit se s běžnými a také moderními metodami potlačování šumu ve zvukových signálech. Jako moderní jsou označovány nové přístupy k tomuto problému využívající fázové informace signálu a tzv. řídké reprezentace signálů. Bylo za úkol navrhnout výpočetní algoritmy rekonstrukce signálů z jejich zašuměného pozorování. Tyto algoritmy následně implementovat v podobě programového kódu a aplikovat na reálné signály. Výstupy rekonstrukčních metod potom měli být porovnány z hlediska výpočetní náročnosti jednotlivých metod a úspěšnosti rekonstrukce na základě objektivního i subjektivního hodnocení.

V práci jsme si jako jednu ze základních metod potlačování šumu představili metodu využívající informaci o magnitudě uložených ve spektrogramu signálu. V rámci této metody jsme si ukázaly dvě nelineární funkce, které se používají pro potlačování malých magnitud signálu. Pro implementaci a následné testování byla použita nelineární funkce $h(x) = x/(x + \eta)$.

Jako moderní metoda byla v práci představena metoda založená na fázi a využívající řídkých reprezentací signálů. Metoda využívá první derivace fáze a skutečnosti, která z ní vyplývá, tedy že pro jednotlivé harmonické komponenty signálu je jejich první derivace v čase po řádcích spektrogramu konstantní. Formulovali jsme minimalizační úlohu vynucující nízkou totální variaci po řádcích (ve směru časové osy) a řídkost výsledného řešení.

V další části práce byla tato moderní metoda rozšířena o druhou derivaci fáze místo první. Tato skutečnost přinesla možnost použít ve výpočtu totální variaci i ve směru frekvenčního rozlišení (tedy po sloupcích), neboť druhá derivace fáze generuje konstantní hodnoty i ve vertikálním směru spektrogramu.

Všechny tři metody potlačování šumu, které jsme si v práci představily, byly implementovány v prostředí MATLAB s využitím toolboxu LTFAT. Metody byly jednotlivě otestovány na stejném vstupním signálu a výsledky okomentovány a zobrazeny jako spektrogram výstupního signálu. Následně jsme provedli komplexnější testování všech tří metod na třech různých vstupních signálech s dvěma různými úrovněmi zašumění. K testovaným signálům byl přidán bílý Gaussův šum s odstupem signálu od šumu 15 dB a 10 dB. Za účelem testování byly nastaveny různé parametry η , λ_1 , λ_{TV} , λ_{TVh} a λ_{TVv} u daných jednotlivých metod. Výsledky jsou zobrazeny v tabulkách. Srovnání úspěšnosti rekonstrukce proběhlo za pomoci objektivní metody SNR a subjektivního názoru po poslechu výstupů jednotlivých metod. Zjistili jsme, že metoda využívající informace uložené v magnitudě nedosahuje tak dobrých výsledků, jako zbylé dvě metody založené na fázi. Avšak oproti předpokladu se nepodařilo prokázat, že by některá z metod založených na fázi byla pro některý typ vstupního signálu výhodnější. Výsledky obou metod jsou si velmi podobné. Naopak z hlediska výpočetní náročnosti je výpočet metody využívající druhou derivaci fáze často výrazně výpočetně náročnější a trvá tedy podstatně déle.

LITERATURA

- [1] RAJMIC, Pavel. *Řídké a nízkohodnostní reprezentace signálů s aplikacemi: Sparse and low-rank representations of signals with applications : zkrácená verze habilitační práce*. Brno: VUTUM, 2015. ISBN 9788021451957.
- [2] ŠPIŘÍK, Jan, Pavel RAJMIC a Vítězslav VESELÝ. Reprezentace signálů: od báží k frámům. *Elektrorevue* [online]. b.r., **2010**(6), 7-8 [cit. 2016-12-07]. ISSN 1213 - 1539. Dostupné z: <http://elektrorevue.cz/cz/download/reprezentace-signalu--od-bazi-k-framum/>
- [3] CHRISTENSEN, Ole. *Frames and bases: an introductory course*. Boston: Birkhäuser, 2008, xviii, 313 s. ISBN 978-0-8176-4677-6.
- [4] PRUSA, Zdenek;balazs. A Noniterative Method for Reconstruction of Phase From STFT Magnitude. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* [online]. IEEE Press, 2017, **25**(5), 1154-1164 [cit. 2018-12-13]. DOI: 10.1109/TASLP.2017.2678166.
- [5] ŠPIŘÍK, Jan. *Modul pro generování "atomů" pro přeparametrizovanou reprezentaci signálu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2010, 52 s.
- [6] PRŮŠA, Zdeněk. *STFT and DGT phase conventions and phase derivatives interpretation* [online]. 2016, , 6 [cit. 2018-12-10]. Dostupné z: <http://lftat.github.io/notes/lftatnote042.pdf>
- [7] HRBÁČEK, Radek, Pavel RAJMIC, Vítězslav VESELÝ a Jan ŠPIŘÍK. Řídké reprezentace signálů: úvod do problematiky. *Elektrorevue* [online]. b.r., **2011**(5), 8 [cit. 2016-12-07]. Dostupné z: <http://www.elektrorevue.cz/cz/download/ridke-reprezentace-signalu--uvod-do-problematiky/>
- [8] Třídy složitosti a Turingovy stroje. *Algoritmy.net* [online]. b.r. [cit. 2016-12-07]. Dostupné z: <https://www.algoritmy.net/article/5774/Tridy-slozitosti>
- [9] CHEN, Ssb, DL DONOHO a MA SAUNDERS. Atomic decomposition by basis pursuit. *Siam Review* [online]. SIAM PUBLICATIONS, 2001, **43**(1), 129-159 [cit. 2018-12-12]. DOI: 10.1137/S003614450037906X. ISSN 0036-1445.
- [10] GERHARD, Rünstler,. *On the design of data sets for forecasting with dynamic factor models* [online]. European Central Bank, 2016, **1893** [cit. 2018-12-12]. Dostupné z: <https://www.ecb.europa.eu/pub/pdf/scpwwps/ecbwp1893.en.pdf>

- [11] MUNRO, Peter R.T. Handbook of Mathematical Methods in Imaging, edited by Otmar Scherzer. *Contemporary Physics* [online]. Routledge, 2013, **54**(1) [cit. 2018-12-12]. DOI: 10.1080/00107514.2012.756938. ISSN 0010-7514.
- [12] CANDES, Emmanuel a Terence TAO. *The Dantzig selector: Statistical estimation when p is much larger than n* [online]. 2005, **35**(6) [cit. 2018-12-12]. DOI: 10.1214/009053606000001523. ISSN 0090-5364.
- [13] MALLAT, S.G. a ZHIFENG ZHANG. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on* [online]. USA: IEEE, 1993, **41**(12), 3397-3415 [cit. 2018-12-12]. DOI: 10.1109/78.258082. ISSN 1053-587X.
- [14] ZHANG, C., X. CAI a Z. SHI. Sparse reconstruction with bat algorithm and orthogonal matching pursuit. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* [online]. Springer Verlag, 2017, **10361**, s. 48-56 [cit. 2018-12-12]. DOI: 10.1007/978-3-319-63309-1_5. ISBN 9783319633084. ISSN 03029743.
- [15] KARAHANOGU, Nazim Burak a Hakan ERDOGAN. A* orthogonal matching pursuit: Best-first search for compressed sensing signal recovery. *Digital Signal Processing* [online]. Elsevier Inc, 2012, **22**(4) [cit. 2018-12-12]. DOI: 10.1016/j.dsp.2012.03.003. ISSN 1051-2004.
- [16] ZÖLZER, Udo. *DAFX: digital audio effects*. Chichester: John Wiley & Sons, 2002, xix, 533 s. ISBN 0-471-49078-4.
- [17] BAYRAM, Ilker. Employing phase information for audio denoising. In: *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on* [online]. IEEE, 2014, s. 2893-2897 [cit. 2018-12-13]. DOI: 10.1109/ICASSP.2014.6854129.
- [18] CONDAT, Laurent. A Generic Proximal Algorithm for Convex Optimization—Application to Total Variation Minimization. *Signal Processing Letters, IEEE* [online]. USA: IEEE, 2014, **21**(8), 985-989 [cit. 2018-12-14]. DOI: 10.1109/LSP.2014.2322123. ISSN 1070-9908. Dostupné z: <https://ieeexplore-ieee-org.ezproxy.lib.vutbr.cz/document/6810809>
- [19] PRŮŠA, Zdeněk, Pavel RAJMÍČ a Nicki HOLINGHAUS. *Employing More Phase Information For Audio Denoising*. 2018, , 11.
- [20] PRŮŠA, Zdeněk, Peter L. SØNDERGAARD, Nicki HOLINGHAUS, Christoph WIESMEYER a Peter BALAZS. *The Large Time-Frequency Analysis Toolbox 2.0* [online]. b.r., s. 419 [cit. 2017-05-10]. DOI: 10.1007/978-3-319-12976-1_25. Dostupné z: http://link.springer.com/10.1007/978-3-319-12976-1_25

- [21] WHITE, Glenn D. a Gary J. LOUIE. *The audio dictionary*. 3rd ed., rev. and expanded. Seattle: University of Washington Press, 2005. ISBN 0-295-98498-8.
- [22] VONDRÁŠEK, Martin. *Odhad SNR řečového signálu snímaného v hlučném prostředí*. České vysoké učení technické v Praze, Fakulta elektrotechnická, 2004. Dostupné také z: http://noel.feld.cvut.cz/speechlab/publications/032_diplomka04.pdf. Vedoucí práce Doc. Ing. Petr Pollák, CSc.

SEZNAM SYMBOLŮ A ZKRATEK

SNR	Signal-to-Noise Ratio, odstup signálu od šumu
LTFAT	The Large Time/Frequency Analysis Toolbox
STFT	krátkodobá Fourierova transformace
DGT	diskrétní Gaborova transformace
pc	fázová konvence
LGD	local group delay, skupinové zpoždění
třídy složitosti NP	nepolynomiální, výsledku je možné dosáhnout v čase vyšším než polynomiálním
BP	Basis Pursuit
LARS	Least Angle Regression
IRLS	Iterative Reweighted Least Squares
FOCUSS	FOCal Underdetermined System Solver
MP	Matching Pursuit
OMP	Orthogonal Matching Pursuit
LASSO	Least Absolute Shrinkage and Selection Operator
TV	totální variace

SEZNAM PŘÍLOH

A Obsah přiloženého CD

A OBSAH PŘÍLOŽENÉHO CD

Příložené CD obsahuje:

- `Pavel_Skyva-diplomova_prace.pdf` – elektronická verze diplomové práce.
- `odsumeniSignalu.zip` – zkomprimovaný soubor obsahující všechny potřebné soubory pro funkci programu v MATLABu včetně testovaných signálů a výstupů testu provedeného v rámci práce.

V souboru `odsumeniSignalu.zip` naleznete všechny funkce popsané v části 6 této práce a také funkce `zobrazDetailMag.m` a `zobrazDetailFaz.m`, které zobrazují detailní informace rekonstrukce jednotlivých metod. Najdeme zde také script `test.m`, který spouští na vybraný vstupní signál se zadaným SNR vstupního signálu a zadanými parametry jednotlivých metod všechny v práci popsané metody potlačování šumu a zobrazuje jejich detailní výsledky. Ve složce `signaly` naleznete všechny testovací signály, které byly při zpracování práce použity. Program byl realizován a spouštěn v programu MATLAB R2013a.

Ve složce `testovani` naleznete audio soubory výsledků testování provedených v části 7.5. Výsledky jsou rozděleny do složek podle názvu vstupního signálu. Jedná-li se o výstup metody využívající magnitudy spektrogramu, soubor je pojmenován:

název vstupního signálu-vstupní SNR-hodnota parametru η -výstupní SNR.wav,

jedná-li se o výstup metody založené na první derivaci fáze, soubor je pojmenován:

název vstupního signálu-vstupní SNR-lambda1-lambdaTV-výstupní SNR.wav,

jedná-li se o výstup metody založené na druhé derivaci fáze, soubor je pojmenován:

název vstupního signálu-vstupní SNR-lambda1-lambdaTVh-lambdaTVv-výstupní SNR.wav.